

1. Program which shows the different data declarations (data types) in COBOL

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM1.

AUTHOR. Indus Computer Services.

*TYPE OF DATA DECLARATION IN COBOL

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NUM    PIC  9    VALUE 1.

77 ALPHA  PIC  A(3) VALUE 'ICS'.

77 ALPHANUM PIC  X(20) VALUE 'WELCOME TO ICS'.

77 DEC    PIC  9V9  VALUE 3.5.

77 SIGNUM PIC  S9   VALUE -5.

PROCEDURE DIVISION.

    DISPLAY 'NUMERIC (9) : ' NUM.

    DISPLAY 'ALPHAPETS (A): ' ALPHA.

    DISPLAY 'ALPHA NUMERIC (X) : ' ALPHANUM.

    DISPLAY 'DECIMAL (V) : ' DEC.

    DISPLAY 'SIGNED VALUE (S) : ' SIGNUM.

    STOP RUN.
```

2. Program which shows the different figurative constants in COBOL

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM2.

AUTHOR. Indus Computer Services.

*TYPES OF FIGURATIVE CONSTANT

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 FZERO    PIC  9(5)  VALUE ZERO.

77 FSPACE   PIC  X(3)  VALUE SPACE.

77 FQUOTE   PIC  X(2)  VALUE QUOTE.

PROCEDURE DIVISION.
```

```
DISPLAY FZERO.  
DISPLAY FSPACE.  
DISPLAY FQUOTE.  
STOP RUN.
```

3. Program to move a data from one variable to the other variable in COBOL

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM3.  
AUTHOR. Indus Computer Services.  
*THE MOVE VERB.  
*TO MOVE THE DATA FROM ONE TO OTHER  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 A    PIC 9  VALUE 5.  
77 B    PIC 9  VALUE ZERO.  
PROCEDURE DIVISION.  
    MOVE A TO B.  
    DISPLAY B.  
    STOP RUN.
```

4. Program to move the numeric data from smaller sending size to larger receiving size

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM4.  
AUTHOR. Indus Computer Services.  
*WHEN THE RECIEVE SIZE IS GREATER.  
*FILLED WITH ZERO IN LEFT FOR NUMERIC  
*FILLED WITH ZERO IN RIGHT FOR DECIMAL  
ENVIRONMENT DIVISION.  
DATA DIVISION.
```

WORKING-STORAGE SECTION.

77 A PIC 999V999 VALUE 345.678.

77 B PIC 9999V9999 VALUE ZERO.

PROCEDURE DIVISION.

MOVE A TO B.

DISPLAY 'RECIEVE SIZE IS GREATER'.

DISPLAY 'FILLED WITH ZERO IN LEFT AND RIGHT : ' B.

STOP RUN.

5. Program to move the numeric data from larger sending size to smaller receiving size

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM5.

AUTHOR. Indus Computer Services.

*WHEN THE RECIEVE SIZE IS SMALLER.

*TRUNCATION OCCURS FROM THE LEFT IN NUMERIC

*TRUNCATION OCCURS FROM THE RIGHT IN DECIMAL

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 99V99 VALUE 34.56.

77 B PIC 9V9 VALUE ZERO.

PROCEDURE DIVISION.

MOVE A TO B.

DISPLAY 'RECIEVE SIZE IS SMALLER'.

DISPLAY 'TRUNCATION OCCURS FROM THE LEFT AND RIGHT : ' B.

STOP RUN.

6. Program to move the alphanumeric data from LARGER sending size to SMALLER receiving size

IDENTIFICATION DIVISION.

```

PROGRAM-ID. PGM6.

AUTHOR. Indus Computer Services.

*WHEN THE RECIEVE SIZE IS SMALLER.

*TRUNCATION OCCURS FROM THE RIGHT IN ALPHA.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC X(5) VALUE 'HELLO'.

77 B PIC X(4) VALUE SPACES.

PROCEDURE DIVISION.

    MOVE A TO B.

    DISPLAY 'RECIEVE SIZE IS SMALLER'.

    DISPLAY 'TRUNCATION OCCURS FROM THE RIGHT : ' B.

    STOP RUN.

```

7. Program to move the alphanumeric data from larger sending size to smaller receiving size

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM7.

AUTHOR. Indus Computer Services.

*WHEN THE RECIEVE SIZE IS GREATER.

*SPACE FILLING OCCURS FROM THE RIGHT IN ALPHANUMERIC.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC X(5) VALUE 'HELLO'.

77 B PIC X(10) VALUE SPACES.

PROCEDURE DIVISION.

    MOVE A TO B.

    DISPLAY 'RECIEVE SIZE IS GREATER'.

    DISPLAY 'SPACE FILLING OCCURS FROM THE RIGHT : ' B B.

    STOP RUN.

```

8. Program to move the data from one to different location

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM8.

AUTHOR. Indus Computer Services.

*MOVE FROM ONE TO DIFFERENT LOCATION

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC X(3) VALUE 'ICS'.
77 B PIC X(4) VALUE SPACES.
77 C PIC X(4) VALUE SPACES.
77 D PIC X(4) VALUE SPACES.

PROCEDURE DIVISION.

    MOVE A TO B,C,D.

    DISPLAY B C D.

    STOP RUN.
```

Arithmetic Verbs : ADD, SUBTRACT, DIVIDE, MULTIPLY, COMPUTE

9. Program to add two numbers

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM9.

AUTHOR. Indus Computer Services.

*ADD TWO NUMBERS

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 12.
77 B PIC 9(2) VALUE 13.

PROCEDURE DIVISION.

    ADD A TO B.
```

```
DISPLAY 'B = A + B ' B.  
STOP RUN.
```

10. Program to add more than two numbers

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM10.  
AUTHOR. Indus Computer Services.  
*ADD MORE THAN TWO NUMBERS  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 A PIC 9(2) VALUE 12.  
77 B PIC 9(2) VALUE 12.  
77 C PIC 9(2) VALUE 12.  
77 D PIC 9(2) VALUE 12.  
PROCEDURE DIVISION.  
ADD A B C TO D.  
DISPLAY 'D = A + B + C +D ' D.  
STOP RUN.
```

11. Program to add more than two numbers and to store in a new one

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM11.  
AUTHOR. Indus Computer Services.  
*ADD MORE THAN TWO NUMBERS STORE IN A NEW DATA NAME  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 A PIC 9(2) VALUE 12.  
77 B PIC 9(2) VALUE 12.
```

```

77 C PIC 9(2) VALUE 12.
77 D PIC 9(2) VALUE 12.
77 E PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

    ADD A B C D GIVING E.

    DISPLAY 'E = A + B + C +D ' E.

    STOP RUN.

```

12. Program to add more than two numbers and to store in multiple data name

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM12.

AUTHOR. Indus Computer Services.

*ADD MORE THAN TWO NUMBERS STORE IN MULTI DATA NAME

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 12.
77 B PIC 9(2) VALUE 12.
77 C PIC 9(2) VALUE 12.
77 D PIC 9(2) VALUE 12.
77 E PIC 9(2) VALUE ZERO.
77 F PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

    ADD A B C D GIVING E F.

    DISPLAY 'E = A + B + C +D ' E.

    DISPLAY 'F = A + B + C +D ' F.

    STOP RUN.

```

13. Program to subtract one number from the other.

```

IDENTIFICATION DIVISION.

```

```

PROGRAM-ID. PGM13.

AUTHOR. Indus Computer Services.

*SUBTRACT TWO NUMBERS

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 24.

77 B PIC 9(2) VALUE 12.

PROCEDURE DIVISION.

    SUBTRACT B FROM A.

    DISPLAY 'A = A-B ' A.

    STOP RUN.

```

14. Program to subtract one number from the other and to store it in a new one.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM14.

AUTHOR. Indus Computer Services.

*SUBTRACT TWO NUMBERS AND TO STORE IT IN A NEW ONE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 24.

77 B PIC 9(2) VALUE 12.

77 C PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

    SUBTRACT B FROM A GIVING C.

    DISPLAY 'C = A-B ' C.

    STOP RUN.

```

15. Program to subtract more than one number from the one and to store it in a new one.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM15.

AUTHOR. Indus Computer Services.

*SUBTRACT TWO NUMBERS AND TO STORE IT IN A NEW ONE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 48.
77 B PIC 9(2) VALUE 12.
77 C PIC 9(2) VALUE 12.
77 D PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

    SUBTRACT B C FROM A GIVING D.

    DISPLAY 'D = A-B-C ' D.

    STOP RUN.

```

16. Program to multiply two numbers.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM16.

AUTHOR. Indus Computer Services.

*MULTIPLY TWO NUMBERS

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 2.
77 B PIC 9(2) VALUE 3.

PROCEDURE DIVISION.

    MULTIPLY A BY B.

    DISPLAY 'B = A * B ' B.

    STOP RUN.

```

17. Program to multiply two numbers and to store it in a new one.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM17.

AUTHOR. Indus Computer Services.

*MULTIPLY TWO NUMBERS AND TO STORE IT IN A NEW ONE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 2.

77 B PIC 9(2) VALUE 3.

77 C PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

MULTIPLY A BY B GIVING C.

DISPLAY 'C = A * B ' C.

STOP RUN.

18. Program to multiply numbers and to store it in multiple values.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM18.

AUTHOR. Indus Computer Services.

*MULTIPLY NUMBERS AND TO STORE IT IN MULTIPLE VALUES

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 2.

77 B PIC 9(2) VALUE 3.

77 C PIC 9(2) VALUE 3.

77 D PIC 9(2) VALUE 3.

77 E PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

```

MULTIPLY A BY B C D.

DISPLAY 'B = A * B ' B.

DISPLAY 'C = A * C ' C.

DISPLAY 'D = A * D ' D.

STOP RUN.

```

19. Program to divide two numbers

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM19.

AUTHOR. Indus Computer Services.

*DIVIDE TWO NUMBERS

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 16.

77 B PIC 9(2) VALUE 2.

PROCEDURE DIVISION.

    DIVIDE B INTO A.

    DISPLAY 'A = A / B ' A.

    STOP RUN.

```

20. Program to divide two numbers and store it in a new one

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM20.

AUTHOR. Indus Computer Services.

*DIVIDE TWO NUMBERS AND TO STORE IT IN NEW ONE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 16.

```

```

77 B PIC 9(2) VALUE 2.
77 C PIC 9(2) VALUE ZERO.
PROCEDURE DIVISION.

    DIVIDE B INTO A GIVING C.
    DISPLAY 'C = A / B ' C.
    STOP RUN.

```

21. Program to divide two numbers and store it in multiple values

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM21.

AUTHOR. Indus Computer Services.

*DIVIDE TWO NUMBERS AND TO STORE IT IN MULTIPLE VALUE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2) VALUE 16.
77 B PIC 9(2) VALUE 2.
77 C PIC 9(2) VALUE ZERO.
77 D PIC 9(2) VALUE ZERO.

PROCEDURE DIVISION.

    DIVIDE B INTO A GIVING C D.
    DISPLAY 'C = A / B ' C.
    DISPLAY 'D = A / B ' D.
    STOP RUN.

```

22. Program to divide two numbers and store quotient and reminder

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM22.

AUTHOR. Indus Computer Services.

```

```

*DIVIDE TWO NUMBERS AND TO STORE QUOTIENT AND REMINDER
ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A   PIC 9(2)   VALUE 15.
77 B   PIC 9(2)   VALUE 2.
77 D   PIC 9(2)   VALUE ZERO.
77 E   PIC 9(2)   VALUE ZERO.

PROCEDURE DIVISION.

    DIVIDE B INTO A GIVING D REMAINDER E.

    DISPLAY 'D = A / B '.

    DISPLAY 'QUOTIENT ' D.

    DISPLAY 'REMINDER ' E.

    STOP RUN.

```

23. Program to display 'Mr' or 'Mrs' based on gender(m/f) using if condition.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM23.

AUTHOR. Indus Computer Services.

*IF CONDITION IN COBOL
ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NAME   PIC A(10)   VALUE SPACES.
77 GENDER PIC X(1)    VALUE SPACES.

PROCEDURE DIVISION.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT NAME.

    DISPLAY 'ENTER YOUR GENDER(M/F) '.

    ACCEPT GENDER.

    *NOTICE THE PERIOD AT THE END OF END-IF

    IF GENDER='M'

```

```

        DISPLAY 'HELLO MR.' NAME
ELSE
        DISPLAY 'HELLO MISS.' NAME
END-IF.

STOP RUN.

```

26. Program that uses nested if statement

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM26.

AUTHOR. Indus Computer Services.

*NESTED IF CONDITION IN COBOL

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NAME  PIC A(10)  VALUE SPACES.
77 AGE   PIC 9(3)   VALUE ZEROS.

PROCEDURE DIVISION.

    DISPLAY 'ENTER YOUR NAME '.
    ACCEPT NAME.

    DISPLAY 'ENTER YOUR AGE '.
    ACCEPT AGE.

    IF AGE = 0
        DISPLAY 'ENTER A VALID AGE > 0'
    ELSE IF AGE < 18
        DISPLAY 'HELLO ' NAME
        DISPLAY 'YOU ARE MINOR'
    ELSE
        DISPLAY 'HELLO ' NAME
        DISPLAY 'YOU ARE MAJOR'
    END-IF.

STOP RUN.

```

27. Program to show the students grades.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM27.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

02 INPUT-DATA.

05 STU-NO PIC 9(3) VALUE ZERO.

05 STU-NAME PIC X(15) VALUE SPACES.

05 TEL PIC 9(3) VALUE ZERO.

05 ENG PIC 9(3) VALUE ZERO.

05 HIN PIC 9(3) VALUE ZERO.

05 MAT PIC 9(3) VALUE ZERO.

05 SCI PIC 9(3) VALUE ZERO.

05 SOC PIC 9(3) VALUE ZERO.

02 OUTPUT-DATA.

05 TOT PIC 9(3) VALUE ZERO.

05 AVG PIC 9(2).9(2) VALUE ZERO.

05 RES PIC X(4) VALUE SPACES.

05 GRA PIC X(6) VALUE SPACES.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER NUMBER'.

ACCEPT STU-NO.

DISPLAY 'ENTE NAME OF THE STUDENT '.

```

ACCEPT STU-NAME.

DISPLAY 'ENTER TEL MARKS'.

ACCEPT TEL.

DISPLAY 'ENTER ENG MARKS'.

ACCEPT ENG.

DISPLAY 'ENTER HIN MARKS'.

ACCEPT HIN.

DISPLAY 'ENTER MAT MARKS'.

ACCEPT MAT.

DISPLAY 'ENTER SCI MARKS'.

ACCEPT SCI.

DISPLAY 'ENTER SOC MARKS'.

ACCEPT SOC.

PROCESS-PARA.

    COMPUTE TOT = TEL + ENG + HIN + MAT + SCI + SOC.

    COMPUTE AVG = TOT / 6.

    IF TEL >= 35 AND ENG >=35  AND HIN >=35 AND MAT >=35
        AND SCI >= 35 AND SOC >= 35 THEN
        MOVE 'PASS' TO RES
    ELSE
        MOVE 'FAIL' TO RES
    END-IF.

    IF AVG >= 60 AND RES = 'PASS' THEN
        MOVE 'FIRST' TO GRA
    ELSE
        IF AVG >= 50 AND AVG < 60 AND RES = 'PASS' THEN
            MOVE 'SECOND' TO GRA
        ELSE
            IF AVG >= 35 AND AVG < 50 AND RES = 'PASS' THEN
                MOVE 'THIRD' TO GRA
            ELSE
                MOVE '*****' TO GRA
            END-IF
        END-IF
    END-IF

```

```

        END-IF.

DISPLAY-PARA.

        DISPLAY 'STU-NO           : '      STU-NO.
        DISPLAY 'STU-NAME        : '      STU-NAME.
        DISPLAY 'TOTAL           : '      TOT.
        DISPLAY 'AVERAGE        : '      AVG.
        DISPLAY 'RESULT          : '      RES.
        DISPLAY 'GRADE           : '      GRA.

```

28. Program using EVALUATE CONDITION with true option.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM28.

AUTHOR. Indus Computer Services.

*PROGRAM CHECK USING EVALUATE CONDITION.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 ENAME    PIC X(10)  VALUE IS SPACES.
77 SEX      PIC X(1)   VALUE IS SPACES.

PROCEDURE DIVISION.

        DISPLAY 'ENTER YOUR NAME'.

        ACCEPT ENAME.

        DISPLAY 'ENTER YOUR SEX(M/F) '.

        ACCEPT SEX.

        EVALUATE TRUE

        WHEN SEX='M'

                DISPLAY 'HELLO MR.' ENAME

        WHEN SEX='F'

                DISPLAY 'HELLO MISS.' ENAME

        WHEN OTHER

                DISPLAY 'PLEASE ENTER A VALID INPUT'

        END-EVALUATE.

```

STOP RUN.

29. Program using EVALUATE option.

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM29.

AUTHOR. Indus Computer Services.

*PROGRAM CHECK USING EVALUATE CONDITION.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 ENAME    PIC X(10)  VALUE IS SPACES.
77 SEX      PIC X(1)   VALUE IS SPACES.

PROCEDURE DIVISION.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT ENAME.

    DISPLAY 'ENTER YOUR SEX(M/F) '.

    ACCEPT SEX.

    EVALUATE SEX
    WHEN 'M'
        DISPLAY 'HELLO MR.' ENAME
    WHEN 'F'
        DISPLAY 'HELLO MISS.' ENAME
    WHEN OTHER
        DISPLAY 'PLEASE ENTER A VALID INPUT'
    END-EVALUATE.

    STOP RUN.
```

30. PROGRAM TO FIND STUDENT'S GRADE USING EVALUATE ALSO.

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM30.
```

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

02 INPUT-DATA.

05 STU-NO PIC 9(3) VALUE ZERO.

05 STU-NAME PIC X(15) VALUE SPACES.

05 TEL PIC 9(3) VALUE ZERO.

05 ENG PIC 9(3) VALUE ZERO.

05 HIN PIC 9(3) VALUE ZERO

05 MAT PIC 9(3) VALUE ZERO.

05 SCI PIC 9(3) VALUE ZERO.

05 SOC PIC 9(3) VALUE ZERO.

02 OUTPUT-DATA.

05 TOT PIC 9(3) VALUE ZERO.

05 AVG PIC 9(2).9(2) VALUE ZERO.

05 RES PIC X(4) VALUE SPACES.

05 GRA PIC X(6) VALUE SPACES..

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER NUMBER'.

ACCEPT STU-NO.

DISPLAY 'ENTE NAME OF THE STUDENT '.

ACCEPT STU-NAME.

DISPLAY 'ENTER TEL MARKS'.

ACCEPT TEL.

DISPLAY 'ENTER ENG MARKS'.

ACCEPT ENG.

```

        DISPLAY 'ENTER HIN MARKS'.
        ACCEPT HIN.
        DISPLAY 'ENTER MAT MARKS'.
        ACCEPT MAT.
        DISPLAY 'ENTER SCI MARKS'.
        ACCEPT SCI.
        DISPLAY 'ENTER SOC MARKS'.
        ACCEPT SOC.

PROCESS-PARA.

        COMPUTE TOT = TEL + ENG + HIN + MAT + SCI + SOC.
        COMPUTE AVG = TOT / 6.
        IF TEL >= 35 AND ENG >=35  AND HIN >=35 AND MAT >=35
            AND SCI >= 35 AND SOC >= 35 THEN
            MOVE 'PASS' TO RES
        ELSE
            MOVE 'FAIL' TO RES
        END-IF.
        EVALUATE TRUE  ALSO TRUE
            WHEN AVG >= 60 ALSO RES = 'PASS'
                MOVE 'FIRST' TO GRA
            WHEN AVG >= 50 AND AVG < 60 ALSO RES = 'PASS'
                MOVE 'SECOND' TO GRA
            WHEN AVG >= 35 AND AVG < 50 ALSO RES = 'PASS'
                MOVE 'THIRD' TO GRA
            WHEN OTHER
                MOVE '*****' TO GRA
        END-EVALUATE.

DISPLAY-PARA.

        DISPLAY 'STU-NO           : ' STU-NO.
        DISPLAY 'STU-NAME        : ' STU-NAME.
        DISPLAY 'TOTAL           : ' TOT.
        DISPLAY 'AVERAGE        : ' AVG.
        DISPLAY 'RESULT          : ' RES.
        DISPLAY 'GRADE           : ' GRA.

```

31. PROGRAM TO FIND STUDENT'S GRADE USING EVALUATE ALSO AND THROUGH

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM31.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

02 INPUT-DATA.

05 STU-NO PIC 9(3).

05 STU-NAME PIC X(15).

05 TEL PIC 9(3).

05 ENG PIC 9(3).

05 HIN PIC 9(3).

05 MAT PIC 9(3).

05 SCI PIC 9(3).

05 SOC PIC 9(3).

02 OUTPUT-DATA.

05 TOT PIC 9(3).

05 AVG PIC 9(2).9(2).

05 RES PIC X(4).

05 GRA PIC X(6).

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER NUMBER'.

ACCEPT STU-NO.

```

    DISPLAY 'ENTE NAME OF THE STUDENT '.
    ACCEPT STU-NAME.
    DISPLAY 'ENTER TEL MARKS'.
    ACCEPT TEL.
    DISPLAY 'ENTER ENG MARKS'.
    ACCEPT ENG.
    DISPLAY 'ENTER HIN MARKS'.
    ACCEPT HIN.
    DISPLAY 'ENTER MAT MARKS'.
    ACCEPT MAT.
    DISPLAY 'ENTER SCI MARKS'.
    ACCEPT SCI.
    DISPLAY 'ENTER SOC MARKS'.
    ACCEPT SOC.

PROCESS-PARA.

    COMPUTE TOT = TEL + ENG + HIN + MAT + SCI + SOC.
    COMPUTE AVG = TOT / 6.
    IF TEL >= 35 AND ENG >=35  AND HIN >=35 AND MAT >=35
        AND SCI >= 35 AND SOC >= 35 THEN
        MOVE 'PASS' TO RES
    ELSE
        MOVE 'FAIL' TO RES
    END-IF.
    EVALUATE AVG  ALSO RES
        WHEN 60 THROUGH 100  ALSO  'PASS'
        MOVE 'FIRST' TO GRA
        WHEN 50 THROUGH 59 ALSO  'PASS'
        MOVE 'SECOND' TO GRA
        WHEN 35 THROUGH 49 ALSO  'PASS'
        MOVE 'THIRD' TO GRA
        WHEN OTHER
            MOVE '*****' TO GRA
    END-EVALUATE.

DISPLAY-PARA.

```

```

        DISPLAY 'STU-NO           : ' STU-NO.
        DISPLAY 'STU-NAME        : ' STU-NAME.
        DISPLAY 'TOTAL           : ' TOT.
        DISPLAY 'AVERAGE        : ' AVG.
        DISPLAY 'RESULT          : ' RES.
        DISPLAY 'GRADE           : ' GRA.

```

32. Program using EVALUATE CONDITION.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM32.

AUTHOR. Indus Computer Services.

*PROGRAM CHECK USING EVALUATE CONDITION.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NOOFDAYS PIC 9(2) VALUE IS ZEROS.

PROCEDURE DIVISION.

    DISPLAY 'NO OF DAYS IN THE MONTH(29/30/31/28)'.

    ACCEPT NOOFDAYS.

    EVALUATE TRUE

    WHEN NOOFDAYS=29 OR NOOFDAYS=28

        DISPLAY 'YOU HAVE ENTERED FEBRUARY'

    WHEN NOOFDAYS=31

        DISPLAY 'YOU HAVE ENTERED ONE OF THE FOLLOWING'
        DISPLAY 'JAN/MAR/MAY/JUL/SEP/NOV/'

    WHEN NOOFDAYS=30

        DISPLAY 'YOU HAVE ENTERED ONE OF THE FOLLOWING'
        DISPLAY 'APR/JUN/AUG/OCT/DEC/'

    WHEN OTHER

        DISPLAY 'ENTER A VALID NUMBER OF DAYS(28/29/30/31)'

    END-EVALUATE.

    STOP RUN.

```

33. Program to use the compound condition (AND)

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM33.

AUTHOR. Indus Computer Services.

*COMPOUND CONDITIONS (AND)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NUM1    PIC 9(2)  VALUE ZEROS.
77 NUM2    PIC 9(2)  VALUE ZEROS.
77 NUM3    PIC 9(2)  VALUE ZEROS.

PROCEDURE DIVISION.

    DISPLAY 'PROGRAM TO FIND THE LARGEST OF THREE NUMBERS '.
    DISPLAY 'ENTER THE FIRST NUMBER'.
    ACCEPT NUM1.
    DISPLAY 'ENTER THE SECOND NUMBER'.
    ACCEPT NUM2.
    DISPLAY 'ENTER THE THIRD NUMBER'.
    ACCEPT NUM3.
    IF NUM1 > NUM2 AND NUM1 > NUM3
        DISPLAY 'NUMBER ' NUM1 ' IS THE LARGEST'
    ELSE IF NUM2 > NUM1 AND NUM2 > NUM3
        DISPLAY 'NUMBER ' NUM2 ' IS THE LARGEST'
    ELSE
        DISPLAY 'NUMBER ' NUM3 ' IS THE LARGEST'
    END-IF.

    STOP RUN.
```

34. Program to use the compound condition (OR)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM34.

AUTHOR. Indus Computer Services.

*COMPOUND CONDITIONS (OR)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NAME    PIC A(10)  VALUE SPACES.

77 GENDER  PIC X(6)   VALUE SPACES.

PROCEDURE DIVISION.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT NAME.

    DISPLAY 'ENTER YOUR GENDER (M (MALE) / F (FEMALE) ' .

    ACCEPT GENDER.

    IF GENDER='M' OR GENDER='MALE'

        DISPLAY 'HELLO MR.' NAME

    ELSE IF GENDER='F' OR GENDER='FEMALE'

        DISPLAY 'HELLO MISS.' NAME

    ELSE

        DISPLAY 'ENTER A VALID GENDER'

    END-IF.

    STOP RUN.

```

35. Program to check the postive/negative/zero number

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM35.

AUTHOR. Indus Computer Services.

*POSTIVE/NEGATIVE/ZERO CLAUSES

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NUM    PIC S9(2)  VALUE ZERO.

```

PROCEDURE DIVISION.

DISPLAY 'ENTER A NUMBER(WITH + OR -) '.

ACCEPT NUM.

IF NUM IS POSITIVE

DISPLAY 'NUMBER ENTERED IS POSITIVE'

ELSE IF NUM IS NEGATIVE

DISPLAY 'NUMBER ENTERED IS NEGATIVE'

ELSE IF NUM IS ZERO

DISPLAY 'NUMBER ENTERED IS ZERO'

END-IF.

STOP RUN.

36. PROGRAM TO DISPLAY THE EDITING CHARACTERS

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM36.

AUTHOR. Indus Computer Services.

*PROGRAM TO DISPLAY THE EDITING CHARACTERS

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 ZNUM PIC ZZ9 VALUE IS ZERO.

77 ASTNUM PIC **9 VALUE IS ZERO.

77 DOLNUM PIC \$99 VALUE IS ZERO.

77 MINNUM PIC -99 VALUE IS ZERO.

77 PLUSNUM PIC +99 VALUE IS ZERO.

77 COMMANUM PIC 9,9 VALUE IS ZERO.

77 BLANKNUM PIC 9B9 VALUE IS ZERO.

77 ZRNUM PIC 990 VALUE IS ZERO.

77 SLASHNUM PIC 9/9 VALUE IS ZERO.

77 BWZNUM PIC Z99 BLANK WHEN ZERO.

PROCEDURE DIVISION.

MOVE 33 TO ZNUM.

```

DISPLAY 'ZERO SUPPRESSED ' ZNUM.
MOVE 33 TO ASTNUM.
DISPLAY 'ASTRISK IN PLACE OF ZERO ' ASTNUM.
MOVE 33 TO DOLNUM.
DISPLAY 'DOLLAR PRECEDING THE NUMERIC ' DOLNUM.
MOVE -33 TO MINNUM.
DISPLAY 'MINUS SIGN PRECEDING THE NUMERIC 'MINNUM.
MOVE 33 TO MINNUM.
DISPLAY 'MINUS SIGN PRECEDING THE NUMERIC 'MINNUM.
MOVE +33 TO PLUSNUM.
DISPLAY 'PLUS SIGN PRECEDING THE NUMERIC 'PLUSNUM.
MOVE -33 TO PLUSNUM.
DISPLAY 'PLUS SIGN PRECEDING THE NUMERIC 'PLUSNUM.
MOVE 33 TO COMMANUM.
DISPLAY 'COMMA SIGN IN THE NUMERIC 'COMMANUM.
MOVE 33 TO BLANKNUM.
DISPLAY 'BLANK IS INSERTED IN THE NUMERIC 'BLANKNUM.
MOVE 33 TO ZRNUM.
DISPLAY 'ZERO IS INSERTED IN THE NUMERIC 'ZRNUM.
MOVE 33 TO SLASHNUM.
DISPLAY 'SLASH IS INSERTED IN THE NUMERIC 'SLASHNUM.
MOVE 033 TO BWZNUM.
DISPLAY 'BLANK INSERTED WHEN ZERO THE NUMERIC 'BWZNUM.
STOP RUN.

```

37. Program with the condition name condition, (88 level number use)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM37.

AUTHOR. Indus Computer Services.

*CONDITION NAME

ENVIRONMENT DIVISION.

DATA DIVISION.

```

WORKING-STORAGE SECTION.

77 MARTIAL-STATUS PIC X(1).

88 MARRIED VALUE 'M'.

88 SINGLE VALUE 'S'.

88 WIDOWED VALUE 'W'.

88 DIVORCED VALUE 'D'.

77 SALARY PIC 9(4) VALUE ZEROS.

77 NAME PIC X(10) VALUE SPACES.

PROCEDURE DIVISION.

DISPLAY 'ENTER YOUR NAME'.

ACCEPT NAME.

DISPLAY 'ENTER YOUR MARTIAL STATUS(M/S/D/W)'.

ACCEPT MARTIAL-STATUS.

IF MARRIED

MOVE 2000 TO SALARY

ELSE IF SINGLE

MOVE 1000 TO SALARY

ELSE IF WIDOWED

MOVE 1500 TO SALARY

ELSE IF DIVORCED

MOVE 1300 TO SALARY

ELSE

MOVE 0 TO SALARY

END-IF.

DISPLAY SALARY.

STOP RUN.

38. Program with the REDEFINE CLAUSE (uses the same location for storage)

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM38.

AUTHOR. Indus Computer Services.

*REDIFINES CLAUSES(SHARES THE SAME LOCATION FOR STORAGE)

```

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 EMP-DETAILS.

    02 EMP-NAME.

        05 NAME    PIC X(5)    VALUE SPACES.

    02 EMP-SALARY  REDEFINES EMP-NAME.

        05 SALARY   PIC X(5).

PROCEDURE DIVISION.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT NAME.

    DISPLAY 'ENTER YOUR SALARY'.

    ACCEPT SALARY.

    DISPLAY 'NAME AND SALARY HAVE THE SAME VALUE'.

    DISPLAY 'NAME :' NAME.

    DISPLAY 'SALARY :' SALARY.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT NAME.

    DISPLAY 'NOW NAME OVERRIDES THE SALARY'.

    DISPLAY 'NAME :' NAME.

    DISPLAY 'SALARY :' SALARY.

    STOP RUN.

```

39. Program with the RENAME CLAUSE (TO REFORM THE GROUP)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM39.

AUTHOR. Indus Computer Services.

*RENAME CLAUSES (TO REFORM THE GROUP)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 EMP.

```

```

02 EMP-DETAILS.

    05 EMP-NO  PIC 9(5)  VALUE ZEROS.

    05 NAME    PIC X(5)  VALUE SPACES.

02 EMP-SAL.

    05 SALARY  PIC X(5).

66 EMP-NAME-SAL RENAME NAME THRU EMP-SAL.

PROCEDURE DIVISION.

    PERFORM ACC-PARA.

    PERFORM DISP-PARA.

    STOP RUN.

DISP-PARA.

    DISPLAY 'EMPLOYEE DETAILS (NAME,NO) '.

    DISPLAY EMP-DETAILS.

    DISPLAY 'EMPLOYEE SALARY (SAL) '.

    DISPLAY EMP-SAL.

    DISPLAY 'EMPLOYEE DETAILS (NAME,SALARY -- REFORMED GROUP) '.

    DISPLAY EMP-NAME-SAL.

ACC-PARA.

    DISPLAY 'ENTER YOUR NAME'.

    ACCEPT NAME.

    DISPLAY 'ENTER YOUR NO'.

    ACCEPT EMP-NO.

    DISPLAY 'ENTER YOUR SALARY'.

    ACCEPT SALARY.

```

40. Program to show the usage clauses.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM40.

AUTHOR. Indus Computer Services.

*USAGE CLAUSES (DISPLAY,COMP,COMP3)

ENVIRONMENT DIVISION.

DATA DIVISION.

```

WORKING-STORAGE SECTION.

77 NUM1 PIC 9(2) USAGE IS DISPLAY.

77 NUM2 PIC 9(2) USAGE IS COMP.

77 NUM3 PIC 9(2) USAGE IS COMP-3.

PROCEDURE DIVISION.

PERFORM ACC-PARA.

PERFORM DISP-PARA.

STOP RUN.

DISP-PARA.

DISPLAY 'ALL THE OUTPUT ARE SAME.ONLY STORAGE IS DIFFERENT'.

DISPLAY 'NUMBER IN DISPLAY(DEFAULT) '.

DISPLAY NUM1.

DISPLAY 'NUMBER IN COMP(BINARY) '.

DISPLAY NUM2.

DISPLAY 'NUMBER IN COMP3(HEXA DECIMAL) '.

DISPLAY NUM3.

ACC-PARA.

DISPLAY 'ENTER NUMBER1'.

ACCEPT NUM1.

DISPLAY 'ENTER NUMBER2'.

ACCEPT NUM2.

DISPLAY 'ENTER NUMBER2'.

ACCEPT NUM3.

41. Program to show the duplicate data names can be fetched (using
qualifier)

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM41.

AUTHOR. Indus Computer Services.

*QUALIFICATION OF DATA NAMES

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

```

01 GRP1.
    02 NUM1    PIC X(5)  VALUE IS 'PRABU'.
01 GRP2.
    02 NUM1    PIC X(5)  VALUE IS 'XXXXX' .
PROCEDURE DIVISION.
    DISPLAY 'NUMBER1 OF GROUP1 ' NUM1 OF GRP1.
    DISPLAY 'NUMBER1 OF GROUP2 ' NUM1 OF GRP2.
    STOP RUN.

```

42. Program to show the move corresponding

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM42.
AUTHOR. Indus Computer Services.
*MOVE CORRESPONDING
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 STU-REC.
    02 NAME    PIC X(5)  VALUE IS 'PRABU'.
    02 NUM    PIC 9(5)   VALUE IS 1234.
01 STU-REP.
    02 NAME    PIC X(5)  VALUE IS SPACES.
    02 FILLER  PIC X(3)  VALUE IS '***'.
    02 NUM    PIC 9(5)  VALUE IS ZEROS.
    02 FILLER  PIC X(3)  VALUE IS '***'.
PROCEDURE DIVISION.
    DISPLAY 'STUDENT RECORD ' STU-REC.
    DISPLAY 'BEFORE MOVE CORR'.
    DISPLAY 'STUDENT REPORT ' STU-REP.
    MOVE CORR STU-REC TO STU-REP.
    DISPLAY 'AFTER MOVE CORR'.
    DISPLAY 'STUDENT REPORT ' STU-REP.

```

STOP RUN.

43. Program to round a value

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM43.

AUTHOR. Indus Computer Services.

*DIVIDE TWO NUMBERS AND TO ROUND

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 A PIC 9(2)V9(3) VALUE 11.
77 B PIC 9(2)V9(3) VALUE 3.
77 C PIC 9(2)V9(2) VALUE ZERO.

PROCEDURE DIVISION.

    DISPLAY 'A IS 11.000'.

    DISPLAY 'B IS 3.000'.

    DIVIDE B INTO A GIVING C ROUNDED.

    DISPLAY 'C = A / B '.

    DISPLAY '3.666 <==> 3.67'.

    DISPLAY C.

    STOP RUN.
```

44. Program to show on size error exceeds.

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM44.

AUTHOR. Indus Computer Services.

*ON SIZE ERROR

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.
```

```

77 A PIC 9(2) VALUE ZERO.
77 B PIC 9(2) VALUE ZERO.
PROCEDURE DIVISION.

    DISPLAY 'ENTER VALUE FOR A '.
    ACCEPT A.

    DISPLAY 'ENTER VALUE FOR B '.
    ACCEPT B.

    ADD A TO B ON SIZE ERROR GO TO ERROR-PARA.

    DISPLAY 'B = A + B ' B.

    STOP RUN.

ERROR-PARA.

    DISPLAY 'SIZE EXCEEDS'.

```

45. Program to use simple perform statement.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM45.

AUTHOR. Indus Computer Services.

*SIMPLE PERFORM

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

    PERFORM DISP-PARA.

    STOP RUN.

DISP-PARA.

    DISPLAY 'THIS IS A SIMPLE PERFORM'.

```

46. Program to use perform statement with number of times.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM46.

```

```
AUTHOR. Indus Computer Services.

*PERFORM WITH NUMBER OF TIMES

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

    PERFORM DISP-PARA 3 TIMES.

    STOP RUN.

DISP-PARA.

    DISPLAY 'THIS IS PERFORM NUMBER OF TIMES'.
```

47. Program to use perform statement with until condition.

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM47.

AUTHOR. Indus Computer Services.

*PERFORM WITH UNTIL CONDITION

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 I    PIC 9    VALUE 1.

PROCEDURE DIVISION.

    PERFORM DISP-PARA UNTIL I > 3.

    STOP RUN.

DISP-PARA.

    COMPUTE I = I + 1.

    DISPLAY 'THIS IS PERFORM UNTIL CONDITION'.
```

48. Program to use perform statement with THRU.

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM48.
```

AUTHOR. Indus Computer Services.

*PERFORM WITH THRU

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 I PIC 9 VALUE ZERO.

PROCEDURE DIVISION.

PERFORM DISP-PARA1 THRU DISP-PARA3.

STOP RUN.

DISP-PARA1.

DISPLAY 'THIS IS PERFORM1'.

DISP-PARA2.

DISPLAY 'THIS IS PERFORM2'.

DISP-PARA3.

DISPLAY 'THIS IS PERFORM3'.

49. Program to use perform statement with varying condition.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM49.

AUTHOR. Indus Computer Services.

*PERFORM WITH VARYING CONDITION

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 I PIC 9 VALUE ZERO.

PROCEDURE DIVISION.

PERFORM DISP-PARA VARYING I FROM 1 BY 1 UNTIL I > 3.

STOP RUN.

DISP-PARA.

DISPLAY 'THIS IS PERFORM WITH VARYING CONDITION'.

50. Program to use perform statement with varying AFTER condition.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM50.

AUTHOR. Indus Computer Services.

*PERFORM WITH VARYING AFTER CONDITION

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 I PIC 9 VALUE ZERO.

77 J PIC 9 VALUE ZERO.

PROCEDURE DIVISION.

DISPLAY 'THIS IS PERFORM WITH VARYING AFTER'.

PERFORM DISP-PARA VARYING I FROM 1 BY 1 UNTIL I > 3

AFTER J FROM 1 BY 1 UNTIL J > 2.

STOP RUN.

DISP-PARA.

DISPLAY '(I,J) ('I','J')'.

51. PROGRAM TO PRINT THE NUMBERS FROM 1 TO GIVEN NUMBER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM51.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

STOP RUN.

```

ACCEPT-PARA.

    DISPLAY 'ENTER N VALUE'.

    ACCEPT N.

PROCESS-PARA.

    MOVE 1 TO I.

    PERFORM UNTIL I > N

        DISPLAY I

        ADD 1 TO I

    END-PERFORM.

```

52. PROGRAM TO FIND THE SUM OF NUMBERS FROM 1 TO GIVEN NUMBER.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM52.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77  N          PIC          9(3)      VALUE ZERO.

77  I          PIC          9(3)      VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

    PERFORM ACCEPT-PARA.

    PERFORM PROCESS-PARA.

    STOP RUN.

ACCEPT-PARA.

    DISPLAY 'ENTER N VALUE'.

    ACCEPT N.

PROCESS-PARA.

    MOVE 1 TO I.

    PERFORM UNTIL I > N

        DISPLAY I

        ADD 1 TO I

```

END-PERFORM.

53. PROGRAM TO FIND THE FACTORIAL OF GIVEN NUMBER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM53.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

77 FACT PIC 9(3) VALUE 1.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

MOVE 1 TO I.

PERFORM UNTIL I > N

COMPUTE FACT = FACT * I

ADD 1 TO I

END-PERFORM.

DISPLAY-PARA.

DISPLAY 'FACT = ' FACT.

54. PROGRAM TO FIND THE SUM OF DIGITS OF GIVEN NUMBER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM54.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

77 REM PIC 9(3) VALUE ZERO.

77 Q PIC 9 VALUE ZERO.

77 TOT PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

PERFORM UNTIL N <= 0

DIVIDE N BY 10 GIVING Q REMAINDER REM

COMPUTE TOT = TOT + REM

DIVIDE N BY 10 GIVING N

END-PERFORM.

DISPLAY-PARA.

DISPLAY 'TOT = ' TOT.

55. PROGRAM TO FIND THE REVERSE OF GIVEN NUMBER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM55.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 REM PIC 9(3) VALUE ZERO.

77 Q PIC 9 VALUE ZERO.

77 REV PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

PERFORM UNTIL N <= 0

DIVIDE N BY 10 GIVING Q REMAINDER REM

COMPUTE REV = REV * 10 + REM

DIVIDE N BY 10 GIVING N

END-PERFORM.

DISPLAY-PARA.

DISPLAY 'REVERSE NUMBER = ' REV.

56. PROGRAM TO FIND WHETHER THE GIVEN NUMBER IS PALINDROME OR NOT.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM56.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 REM PIC 9(3) VALUE ZERO.

77 Q PIC 9 VALUE ZERO.

77 REV PIC 9(3) VALUE ZERO.

77 P PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

MOVE N TO P.

PROCESS-PARA.

PERFORM UNTIL N <= 0

DIVIDE N BY 10 GIVING Q REMAINDER REM

COMPUTE REV = REV * 10 + REM

DIVIDE N BY 10 GIVING N

END-PERFORM.

DISPLAY-PARA.

IF REV = P THEN

DISPLAY 'GIVEN NUMBER IS PALINDROME '

ELSE

DISPLAY 'GIVEN NUMBER IS NOT PALINDROME '

END-IF.

57. PROGRAM TO FIND THE GIVEN NUMBER IS PRIME OR NOT.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM57.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

77 REM PIC 9(3) VALUE ZERO.

77 Q PIC 9 VALUE ZERO.

77 OPTION PIC X(3) VALUE 'YES'.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

MOVE 2 TO I.

PERFORM UNTIL I >= N

DIVIDE N BY I GIVING Q REMAINDER REM

IF REM = 0 THEN

DISPLAY 'GIVEN NUMBER IS NOT PRIME'

STOP RUN

END-IF

ADD 1 TO I

END-PERFORM.

DISPLAY-PARA.

IF N = I THEN

DISPLAY 'GIVEN NUMBER IS PRIME'

END-IF.

58. PROGRAM TO PRINT PRIME NUMBERS FROM 1 TO GIVEN NUMBER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM58.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

77 K PIC 9(3) VALUE ZERO.

77 REM PIC 9(3) VALUE ZERO.

77 Q PIC 9 VALUE ZERO.

77 OPTION PIC X(3) VALUE 'YES'.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

PERFORM VARYING K FROM 1 BY 1 UNTIL K > N

MOVE 2 TO I

PERFORM UNTIL I >= K OR OPTION = 'NO'

DIVIDE K BY I GIVING Q REMAINDER REM

IF REM = 0 THEN

MOVE 'NO' TO OPTION

END-IF

ADD 1 TO I

END-PERFORM

IF K = I THEN

```
        DISPLAY K
    END-IF
    MOVE 'YES' TO OPTION
END-PERFORM.
```

59.PROGRAM TO PRINT ODD NUMBERS FROM 1 TO GIVEN NUMBER.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PGM59.
AUTHOR. Indus Computer Services.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77  N          PIC          9(3) VALUE  ZERO.
77  I          PIC          9(3) VALUE  ZERO.
PROCEDURE DIVISION.
MAIN-PARA.
    PERFORM ACCEPT-PARA.
    PERFORM PROCESS-PARA VARYING I FROM 1 BY 2 UNTIL I > N.
    STOP RUN.
ACCEPT-PARA.
    DISPLAY 'ENTER N VALUE'.
    ACCEPT N.
PROCESS-PARA.
    DISPLAY I.
```

60 . PROGRAM TO PRINT EVEN NUMBERS FROM 2 TO GIVEN NUMBER.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PGM60.
AUTHOR. Indus Computer Services.
ENVIRONMENT DIVISION.
```

DATA DIVISION.

WORKING-STORAGE SECTION.

77 N PIC 9(3) VALUE ZERO.

77 I PIC 9(3) VALUE ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA.

PERFORM PROCESS-PARA VARYING I FROM 2 BY 2 UNTIL I > N.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER N VALUE'.

ACCEPT N.

PROCESS-PARA.

DISPLAY I.

61. Program USING NEXT SENTENCE.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM61.

AUTHOR. Indus Computer Services.

*PROGRAM USING NEXT SENTENCE.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 NUM PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

DISPLAY 'ENTER THE VALUE 999 TO SKIP THE IF(NEXT SENTENCE)'.

ACCEPT NUM.

IF NUM=999

NEXT SENTENCE

ELSE

DISPLAY 'NOT IN NEXT SENTENCE'

END-IF.

```
DISPLAY 'NEXT SENTENCE'.  
STOP RUN.
```

62. Program USING CONTINUE.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM66.  
AUTHOR. Indus Computer Services.  
*PROGRAM USING CONTINUE  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
77 NUM    PIC 9(3)  VALUE IS ZERO.  
PROCEDURE DIVISION.  
    DISPLAY 'ENTER THE VALUE 999 TO SKIP AND CONTINUE WITH NEXT'.  
    ACCEPT NUM.  
    IF NUM=999  
        CONTINUE  
        DISPLAY 'IN CONTINUE'  
    ELSE  
        DISPLAY 'NOT IN CONTINUE'.  
    DISPLAY 'OUT OF LOOP'.  
    STOP RUN.
```

63. Program that show the table with subscript.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM63.  
AUTHOR. Indus Computer Services.  
*ONE DIMENSIONAL TABLE WITH SUBSCRIPT  
ENVIRONMENT DIVISION.  
DATA DIVISION.
```

WORKING-STORAGE SECTION.

01 STU-REC.

02 STU-DETAILS OCCURS 3 TIMES.

05 STU-NO PIC 9(5) VALUE ZEROS.

05 STU-NAME PIC X(10) VALUE SPACES.

77 I PIC 9 VALUE 1.

PROCEDURE DIVISION.

PERFORM ACC-PARA 3 TIMES.

COMPUTE I = 1.

PERFORM DISP-PARA 3 TIMES.

STOP RUN.

ACC-PARA.

DISPLAY 'ENTER THE STUDENT NUMBER'.

ACCEPT STU-NO(I).

DISPLAY 'ENTER THE STUDENT NAME'.

ACCEPT STU-NAME(I).

COMPUTE I = I + 1.

DISP-PARA.

DISPLAY 'STUDENT NUMBER ' STU-NO(I).

DISPLAY 'STUDENT NAME ' STU-NAME(I).

COMPUTE I = I + 1.

64. PROGRAM TO FIND AVERAGE OF LIST OF NUMBERS USING OCCURS.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM64.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 SINGLE-DIM.

05 A PIC 9(2) OCCURS 5 TIMES.

77 I PIC 9 VALUE ZERO.

77 TOT PIC 9(3) VALUE ZERO.

```

77  AVG      PIC      9(2)V99  VALUE  ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

        PERFORM ACCEPT-PARA VARYING I FROM 1 BY 1 UNTIL I > 5.
        PERFORM PROCESS-PARA VARYING I FROM 1 BY 1 UNTIL I > 5.
        PERFORM DISPLAY-PARA.
        STOP RUN.

ACCEPT-PARA.

        DISPLAY 'ENTER NUMER ' I.
        ACCEPT A(I) .

PROCESS-PARA.

        COMPUTE TOT = TOT + A(I) .

DISPLAY-PARA.

        DISPLAY 'ENTERED VALUES ARE : ' .
        PERFORM VARYING I FROM 1 BY 1 UNTIL I > 5
                DISPLAY A(I)
        END-PERFORM.
        DISPLAY 'TOTAL                : ' TOT.
        COMPUTE AVG = TOT / 5.
        DISPLAY 'AVERAGE              : ' AVG.

```

65. PROGRAM TO FIND THE MAXIMUM NUMBER FROM LIST OF VALUES.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM65.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01  SINGLE-DIM.

        05  A      PIC      9(2) OCCURS 5 TIMES.

77  I          PIC      9 VALUE  ZERO..

77  MAX        PIC      9(3) VALUE  ZERO.

```

PROCEDURE DIVISION.

MAIN-PARA.

PERFORM ACCEPT-PARA VARYING I FROM 1 BY 1 UNTIL I > 5.

MOVE A(1) TO MAX.

PERFORM PROCESS-PARA VARYING I FROM 2 BY 1 UNTIL I > 5.

PERFORM DISPLAY-PARA.

STOP RUN.

ACCEPT-PARA.

DISPLAY 'ENTER NUMER ' I.

ACCEPT A(I).

PROCESS-PARA.

IF MAX < A(I) THEN

MOVE A(I) TO MAX

END-IF.

DISPLAY-PARA.

DISPLAY 'MAXIMUM NUMBER = ' MAX.

66. PROGRAM TO ARRANGE THE NUMBERS IN ASCENDING ORDER.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM66.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 SINGLE-DIM.

05 A PIC 9(2) OCCURS 5 TIMES.

77 I PIC 9 VALUE ZERO.

77 J PIC 9 VALUE ZERO.

77 K PIC 9 VALUE ZERO.

77 TEMP PIC 9(3).

PROCEDURE DIVISION.

MAIN-PARA.

```

        PERFORM ACCEPT-PARA VARYING I FROM 1 BY 1 UNTIL I > 5.
        PERFORM PROCESS-PARA.
        DISPLAY 'ASCENDING ORDER  : '.
        PERFORM DISPLAY-PARA VARYING I FROM 1 BY 1 UNTIL I > 5.
        STOP RUN.

ACCEPT-PARA.
        DISPLAY 'ENTER NUMER ' I.
        ACCEPT A(I).

PROCESS-PARA.
        PERFORM VARYING I FROM 1 BY 1 UNTIL I > 4
        COMPUTE K = I + 1
        PERFORM VARYING J FROM K BY 1 UNTIL J > 5
                IF A(I) > A(J) THEN
                        MOVE A(I) TO TEMP
                        MOVE A(J) TO A(I)
                        MOVE TEMP TO A(J)
                END-IF
        END-PERFORM
END-PERFORM.

DISPLAY-PARA.
        DISPLAY A(I).

```

67. PROGRAM TO FIND THE ADDITION OF TWO MATRICES.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM67.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01  MATRIX-A.

    05  A1      OCCURS  2  TIMES.

        10  A          OCCURS  2  TIMES          PIC  9(2).

```

```

01  MATRIX-B.
      05  B1      OCCURS  2  TIMES.
          10  B      OCCURS  2  TIMES      PIC  9(2) .
01  MATRIX-C.
      05  C1      OCCURS  2  TIMES.
          10  C      OCCURS  2  TIMES      PIC  9(2) .
77  I          PIC      9 VALUE  ZERO.
77  J          PIC      9 VALUE  ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

      PERFORM READ-MATRIX-A.

      PERFORM READ-MATRIX-B.

      PERFORM PROCESS-PARA.

      PERFORM DISPLAY-PARA.

      STOP RUN.

READ-MATRIX-A.

      PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
          PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
              DISPLAY 'ENTER VALUE  FOR MATRIX A '
              ACCEPT A(I, J)
          END-PERFORM
      END-PERFORM.

READ-MATRIX-B.

      PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
          PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
              DISPLAY 'ENTER VALUE  FOR MATRIX B '
              ACCEPT B(I, J)
          END-PERFORM
      END-PERFORM.

PROCESS-PARA.

      PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
          PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
              COMPUTE C(I, J) = A(I, J) + B(I, J)
          END-PERFORM

```

```

        END-PERFORM.

DISPLAY-PARA.

        PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2

                PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2

                        DISPLAY C(I, J)

                END-PERFORM

        END-PERFORM.

```

68. PROGRAM TO FIND THE MULTIPLICATION OF TWO MATRICES.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM68.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01  MATRIX-A.

    05  A1      OCCURS  2  TIMES.

        10  A      OCCURS  2  TIMES          PIC  9(2) .

01  MATRIX-B.

    05  B1      OCCURS  2  TIMES.

        10  B      OCCURS  2  TIMES          PIC  9(2) .

01  MATRIX-C.

    05  C1      OCCURS  2  TIMES.

        10  C      OCCURS  2  TIMES          PIC  9(2) .

77  I          PIC      9 VALUE  ZERO.

77  J          PIC      9 VALUE  ZERO.

77  K          PIC      9 VALUE  ZERO.

PROCEDURE DIVISION.

MAIN-PARA.

        PERFORM READ-MATRIX-A.

        PERFORM READ-MATRIX-B.

        PERFORM PROCESS-PARA.

```

```

        PERFORM DISPLAY-PARA.

    STOP RUN.

READ-MATRIX-A.

    PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
        PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
            DISPLAY 'ENTER VALUE  FOR MATRIX A '
            ACCEPT A(I, J)
        END-PERFORM
    END-PERFORM.

READ-MATRIX-B.

    PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
        PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
            DISPLAY 'ENTER VALUE  FOR MATRIX B '
            ACCEPT B(I, J)
        END-PERFORM
    END-PERFORM.

PROCESS-PARA.

    PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
        PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
            MOVE ZERO TO C(I, J)
            PERFORM VARYING K FROM 1 BY 1 UNTIL K > 2
                COMPUTE C(I, J) = C(I, J)  + A(I, K) * B(K, J)
            END-PERFORM
        END-PERFORM
    END-PERFORM.

DISPLAY-PARA.

    PERFORM VARYING I FROM 1 BY 1 UNTIL I > 2
        PERFORM VARYING J FROM 1 BY 1 UNTIL J > 2
            DISPLAY C(I, J)
        END-PERFORM
    END-PERFORM.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM69.
AUTHOR. Indus Computer Services.
*TWO DIMENSIONAL TABLE
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.

01 STUD-TABLE.

02 STUD-REC OCCURS 2 TIMES.

03 STUD-NAME PIC X(5) VALUE SPACES.

03 STUD-NUMBER PIC X(3) VALUE SPACES.

03 MARKS OCCURS 3 TIMES.

04 MARK PIC 9(3) VALUE ZEROS.

77 I PIC 9(2) VALUE ZEROS.

77 J PIC 9(2) VALUE ZEROS.

PROCEDURE DIVISION.

PERFORM ACCEPT-DETAILS VARYING I FROM 1 BY 1 UNTIL I > 2.

PERFORM DISP-DETAILS VARYING I FROM 1 BY 1 UNTIL I > 2.

STOP RUN.

ACCEPT-DETAILS.

DISPLAY 'ENTER THE STUDENT NAME'.

ACCEPT STUD-NAME(I).

DISPLAY 'ENTER THE STUDENT NUMBER'.

ACCEPT STUD-NUMBER(I).

DISPLAY 'ENTER THE MARKS'.

PERFORM ACCEPT-MARK VARYING J FROM 1 BY 1 UNTIL J > 3.

ACCEPT-MARK.

ACCEPT MARK(I,J).

DISP-DETAILS.

DISPLAY '*****'.

DISPLAY STUD-NAME(I)

DISPLAY 'STUDENT NUMBER ' STUD-NUMBER(I).

```

        PERFORM DISP-MARK VARYING J FROM 1 BY 1 UNTIL J > 3.
DISP-MARK.

        DISPLAY 'MARK 'J ' ' MARK(I,J).

```

70. Program that show the table with INDEX.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM70.

AUTHOR. Indus Computer Services.

*ONE DIMENSIONAL TABLE WITH INDEX

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

    02 STU-DETAILS OCCURS 2 TIMES INDEXED BY A1.

        05 STU-NO    PIC 9(5)  VALUE ZEROS.

        05 STU-NAME  PIC X(10)  VALUE SPACES.

PROCEDURE DIVISION.

    PERFORM ACC-PARA 2 TIMES.

    PERFORM DISP-PARA 2 TIMES.

    STOP RUN.

ACC-PARA.

    DISPLAY 'ENTER THE STUDENT NUMBER'.

    ACCEPT STU-NO(A1).

    DISPLAY 'ENTER THE STUDENT NAME'.

    ACCEPT STU-NAME(A1).

DISP-PARA.

    SET A1 TO 1.

    DISPLAY 'STUDENT NUMBER ' STU-NO(A1).

    DISPLAY 'STUDENT NAME ' STU-NAME(A1).

```

71. Program that search the table for a record (sequential search) .

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM71.

AUTHOR. Indus Computer Services.

*PROGRAM TO SEARCH A RECORD FROM A TABLE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

02 STU-DETAILS OCCURS 2 TIMES INDEXED BY A1.

05 STU-NO PIC 9(5) VALUE ZEROS.

05 STU-NAME PIC X(10) VALUE SPACES.

77 IN-NO PIC 9(5) VALUE IS ZERO.

77 I PIC 9 VALUE IS 1.

PROCEDURE DIVISION.

PERFORM ACC-PARA 2 TIMES.

PERFORM SER-PARA.

ACC-PARA.

DISPLAY 'ENTER THE STUDENT NUMBER'.

ACCEPT STU-NO(I).

DISPLAY 'ENTER THE STUDENT NAME'.

ACCEPT STU-NAME(I).

COMPUTE I =I + 1.

SER-PARA.

DISPLAY 'ENTER THE NUMBER TO SEARCH'.

ACCEPT IN-NO.

SET A1 TO 1.

SEARCH STU-DETAILS

AT END GO TO PARA-NO

WHEN IN-NO = STU-NO(A1)

GO TO DISP-PARA

END-SEARCH.

DISP-PARA.

DISPLAY 'STUDENT NUMBER ' STU-NO(A1) .

```

        DISPLAY 'STUDENT NAME ' STU-NAME(A1) .
        PERFORM END-RTN.
PARA-NO.
        DISPLAY 'NOT FOUND'.
        PERFORM END-RTN.
END-RTN.
STOP RUN.

```

72. Program that search the table for a record (binary search) SEARCH ALL.

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM72.
AUTHOR. Indus Computer Services.
*PROGRAM TO SEARCH A RECORD FROM A TABLE (BINARY)
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 STU-REC.
    02 STU-DETAILS OCCURS 3 TIMES
        ASCENDING KEY IS STU-NO
        INDEXED BY A1.
    05 STU-NO    PIC 9(2)  VALUE ZEROS.
    05 STU-NAME  PIC X(10) VALUE SPACES.
77 IN-NO    PIC 9(2)  VALUE IS ZERO.
77 I      PIC 9    VALUE IS 1.
PROCEDURE DIVISION.
    DISPLAY 'ENTER THE STUDENT NUMBER IN ASCENDING ORDER'.
    PERFORM ACC-PARA 3 TIMES.
    PERFORM SER-PARA.
ACC-PARA.
    DISPLAY 'ENTER THE STUDENT NUMBER'.
    ACCEPT STU-NO(I).
    DISPLAY 'ENTER THE STUDENT NAME'.

```

```

ACCEPT STU-NAME(I) .

COMPUTE I =I + 1.

SER-PARA.

DISPLAY 'ENTER THE NUMBER TO SEARCH'.

ACCEPT IN-NO.

SEARCH ALL STU-DETAILS

        AT END GO TO PARA-NO

        WHEN STU-NO(A1) = IN-NO

        GO TO DISP-PARA

END-SEARCH.

DISP-PARA.

        DISPLAY 'STUDENT NUMBER ' STU-NO(A1) .

        DISPLAY 'STUDENT NAME ' STU-NAME(A1) .

        PERFORM END-RTN.

PARA-NO.

        DISPLAY 'NOT FOUND'.

        PERFORM END-RTN.

END-RTN.

STOP RUN.

```

73. Program that create the table on depending clause

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM73.

AUTHOR. Indus Computer Services.

*PROGRAM TO CEREATE TABLE ON DYNAMIC VALUE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 STU-REC.

    02 STU-DETAILS OCCURS 3 TO 5 TIMES DEPENDING ON J.

        05 STU-NO    PIC 9(2)  VALUE ZEROS.

        05 STU-NAME  PIC X(10) VALUE SPACES.

```

```

77 I      PIC 9  VALUE IS 1.
77 J      PIC 9  VALUE IS ZERO.
PROCEDURE DIVISION.

    MOVE '4' TO J.

    PERFORM ACC-PARA J TIMES.

    COMPUTE I = 1.

    PERFORM DISP-PARA J TIMES.

    PERFORM END-RTN.

ACC-PARA.

    DISPLAY 'ENTER THE STUDENT NUMBER'.

    ACCEPT STU-NO(I).

    DISPLAY 'ENTER THE STUDENT NAME'.

    ACCEPT STU-NAME(I).

    COMPUTE I =I + 1.

DISP-PARA.

    DISPLAY 'STUDENT NUMBER ' STU-NO(I).

    DISPLAY 'STUDENT NAME ' STU-NAME(I).

    COMPUTE I = I + 1.

END-RTN.

    STOP RUN.

```

74. Program to write in to a sequential file

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM74.

AUTHOR. Indus Computer Services.

*PROGRAM TO WRITE TO A SEQUENTIAL FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT WRITE-FILE ASSIGN TO DD1

        ORGANIZATION IS SEQUENTIAL

        ACCESS MODE IS SEQUENTIAL.

```

DATA DIVISION.

FILE SECTION.

FD WRITE-FILE.

01 EMP-REC.

02 EMP-NO PIC 9(3).

02 EMP-NAME PIC X(10).

02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

01 IN-REC.

02 IN-NO PIC 9(3).

02 IN-NAME PIC X(10).

02 IN-SAL PIC 9(5).

77 OPTION PIC X(3) VALUE IS 'YES'.

PROCEDURE DIVISION.

PERFORM OPEN-FILE.

PERFORM WRITE-TO-FILE.

PERFORM CLOSE-FILE.

STOP RUN.

OPEN-FILE.

OPEN OUTPUT WRITE-FILE.

CLOSE-FILE.

CLOSE WRITE-FILE.

WRITE-TO-FILE.

PERFORM ACC-INPUT UNTIL OPTION='NO'.

ACC-INPUT.

DISPLAY 'ENTER THE EMPLOYEE NUMBER'.

ACCEPT IN-NO.

DISPLAY 'ENTER THE EMPLOYEE NAME'.

ACCEPT IN-NAME.

DISPLAY 'ENTER THE EMPLOYEE SALARY'.

ACCEPT IN-SAL.

MOVE IN-REC TO EMP-REC

WRITE EMP-REC.

DISPLAY 'DO YOU WANT TO WRITE MORE RECORDS(YES/NO)'.

ACCEPT OPTION.

75. Program to Read from a sequential file

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM75.

AUTHOR. Indus Computer Services.

*PROGRAM TO READ FROM A SEQUENTIAL FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

 SELECT READ-FILE ASSIGN TO DD1

 ORGANIZATION IS SEQUENTIAL

 ACCESS MODE IS SEQUENTIAL.

DATA DIVISION.

FILE SECTION.

FD READ-FILE.

01 EMP-REC.

 02 EMP-NO PIC 9(3).

 02 EMP-NAME PIC X(10).

 02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

01 OUT-REC.

 02 OUT-NO PIC 9(3).

 02 OUT-NAME PIC X(10).

 02 OUT-SAL PIC 9(5).

77 OPTION PIC X(3) VALUE IS 'YES'.

PROCEDURE DIVISION.

 PERFORM OPEN-FILE.

 PERFORM READ-PARA UNTIL OPTION = 'NO'.

 PERFORM CLOSE-FILE.

 STOP RUN.

OPEN-FILE.

```

        OPEN INPUT READ-FILE.
CLOSE-FILE.
        CLOSE READ-FILE.
READ-PARA.
        READ READ-FILE
            AT END
MOVE 'NO' TO OPTION
        NOT AT END
        MOVE EMP-REC TO OUT-REC
        DISPLAY OUT-REC.
END-READ.

```

76. REPORT PROGRAMS

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM76.
AUTHOR. Indus Computer Services.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT EMPLOYEE-DATA ASSIGN TO EMP-DAT.
        SELECT PAYROLL-LISTING ASSIGN TO PRI-DAT.
DATA DIVISION.
FILE SECTION.
FD EMPLOYEE-DATA
        LABEL RECORDS ARE STANDARD.
01 EMPLOYEE-RECORD.
        05 EMPLOYEE-NAME-IN          PIC      X(20) .
        05 HOURS-WORKED-IN           PIC      9(2) .
        05 HOURLY-RATE-IN             PIC      9V99.
FD PAYROLL-LISTING
        LABEL RECORDS ARE OMITTED.
01 PRINT-REC.

```

```

05                                     PIC      X(20) .
05  NAME-OUT                         PIC      X(20) .
05                                     PIC      X(10) .
05  HOURS-OUT                        PIC      9(2) .
05                                     PIC      X(8) .
05  RATE-OUT                         PIC      9.99.
05                                     PIC      X(6) .
05  WEEKLY-WAGES-OUT                PIC      999.99.

WORKING-STORAGE SECTION.

01  ARE-THERE-MORE-RECORDS          PIC      X(3) VALUE 'YES'.

PROCEDURE DIVISION.

100-MAIN-MODULE.

    OPEN INPUT EMPLOYEE-DATA
        OUTPUT PAYROLL-LISTING.

    PERFORM UNTIL  ARE-THERE-MORE-RECORDS = 'NO'
        READ EMPLOYEE-DATA
            AT END
                MOVE 'NO' TO ARE-THERE-MORE-RECORDS
            NOT AT END
                PERFORM 200-WAGE-ROUTINE
        END-READ
    END-PERFORM.

    CLOSE EMPLOYEE-DATA
        PAYROLL-LISTING.

    STOP RUN.

200-WAGE-ROUTINE.

    MOVE SPACES TO PRINT-REC.

    MOVE EMPLOYEE-NAME-IN TO NAME-OUT.

    MOVE HOURS-WORKED-IN TO HOURS-OUT.

    MOVE HOURLY-RATE-IN TO RATE-OUT.

    MULTIPLY HOURS-WORKED-IN BY HOURLY-RATE-IN
        GIVING WEEKLY-WAGES-OUT

    WRITE PRINT-REC.

```

77. REPORT PROGRAM 2

IDENTIFICATION DIVISION.

PROGRAM-ID.PGM77.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT IN-EMPLOYEE-FILE ASSIGN TO EMPDAT1

ORGANIZATION IS SEQUENTIAL .

SELECT OUT-REPORT-FILE ASSIGN TO PRINTER1.

ORGANIZATION IS SEQUENTIAL .

DATA DIVISION.

FILE SECTION.

FD IN-EMPLOYEE-FILE

LABEL RECORDS ARE STANDARD.

01 IN-EMPLOYEE-REC.

05 IN-EMPLOYEE-NO PIC X(3).

05 IN-EMPLOYEE-LAST-NAME PIC X(20).

05 IN-EMPLOYEE-FIRST-NAME PIC X(10).

05 IN-RANK PIC XX.

05 IN-SALARY PIC 9(6)V99.

FD OUT-REPORT-FILE

LABEL RECORDS ARE OMITTED.

01 OUT-REPORT-REC PIC X(132).

WORKING-STORAGE SECTION.

01 WS-WORK-AREAS.

05 ARE-THERE-MORE-RECORDS PIC X(3) VALUE 'YES'.

05 WS-PROFESSOR-CTR PIC 9(3) VALUE ZEROS.

05 WS-ASSOCIATE-CTR PIC 9(3) VALUE ZEROS.

05 WS-ASSISTANT-CTR PIC 9(3) VALUE ZEROS.

05 WS-INSTRUCTOR-CTR PIC 9(3) VALUE ZEROS.

05 WS-PROFESSOR-COST PIC 9(7)V99 VALUE ZEROS.

05	WS-ASSOCIATE-COST	PIC	9(7)V99	VALUE ZEROS.
05	WS-ASSISTANT-COST	PIC	9(7)V99	VALUE ZEROS.
05	WS-INSTRUCTOR-COST	PIC	9(7)V99	VALUE ZEROS.
05	WS-TOTAL-COST	PIC	9(9)V99	VALUE ZEROS.
05	NEW-SAL	PIC	9(7)V99	VALUE ZEROS.
01 HL-HEADER-1.				
05		PIC	X(49)	VALUE SPACES.
05		PIC	X(25)	
	VALUE			'UNIVERSITY PAYROLL REPORT'.
05		PIC	X(58)	VALUE SPACES.
01 HL-HEADER-2.				
05		PIC	X(24)	VALUE SPACES.
05		PIC	X(30)	VALUE 'RANK'.
05		PIC	X(20)	
	VALUE			'NO OF EMPLOYEES'.
05		PIC	X(25)	
	VALUE			'COST OF PROPOSED INCREASE'.
05		PIC	X(33)	VALUE SPACES.
01 TL-TOTAL-LINE.				
05		PIC	X(24)	VALUE SPACES.
05	TL-RANK	PIC	X(10).	
05		PIC	X(26)	VALUE SPACES.
05	TL-NO-OF-EMPLOYEES	PIC	9(3).	
05		PIC	X(16)	VALUE SPACES.
05	TL-COST	PIC	\$Z,ZZZ,ZZ9.99.	
05		PIC	X(41)	VALUE SPACES.
01 TL-FINAL-TOTAL-LINE.				
05		PIC	X(32)	VALUE SPACES.
05		PIC	X(46)	
	VALUE			'TOTAL UNIVERSITY BUGGEST WILL BE INCREASED BY'.
05	TL-TOTAL-COST	PIC	\$Z,ZZZ,ZZ9.99.	
05		PIC	X(39)	VALUE SPACES.

PROCEDURE DIVISION.

100-MAIN-MODULE.

```

OPEN      INPUT  IN-EMPLOYEE-FILE
          OUTPUT OUT-REPORT-FILE.
PERFORM UNTIL ARE-THERE-MORE-RECORDS = 'NO'
    READ IN-EMPLOYEE-FILE
    AT END
        MOVE 'NO' TO ARE-THERE-MORE-RECORDS
    NOT AT END
        PERFORM 200-CALC-RTN
    END-READ
END-PERFORM.
PERFORM 300-FINAL-RTN
CLOSE     IN-EMPLOYEE-FILE
          OUT-REPORT-FILE.
STOP RUN.

200-CALC-RTN.
    IF IN-RANK = 'FP'
        MULTIPLY IN-SALARY BY .062 GIVING NEW-SAL
        ADD NEW-SAL TO WS-PROFESSOR-COST
        ADD 1 TO WS-PROFESSOR-CTR
    END-IF.
    IF IN-RANK = 'AS'
        MULTIPLY IN-SALARY BY .081 GIVING NEW-SAL
        ADD NEW-SAL TO WS-ASSOCIATE-COST
        ADD 1 TO WS-ASSOCIATE-CTR
    END-IF.
    IF IN-RANK = 'AP'
        MULTIPLY IN-SALARY BY .083 GIVING NEW-SAL
        ADD NEW-SAL TO WS-ASSISTANT-COST
        ADD 1 TO WS-ASSISTANT-CTR
    END-IF.
    IF IN-RANK = 'AP'
        MULTIPLY IN-SALARY BY .102 GIVING NEW-SAL
        ADD NEW-SAL TO WS-INSTRUCTOR-COST
        ADD 1 TO WS-INSTRUCTOR-CTR

```

```

END-IF.

300-FINAL-RTN.

WRITE OUT-REPORT-REC FROM HL-HEADER-1
      AFTER ADVANCING PAGE.

WRITE OUT-REPORT-REC FROM HL-HEADER-2
      AFTER ADVANCING 5 LINES.

MOVE 'FULL' TO TL-RANK.

MOVE WS-PROFESSOR-CTR TO TL-NO-OF-EMPLOYEES.

MOVE WS-PROFESSOR-COST TO TL-COST.

WRITE OUT-REPORT-REC FROM TL-TOTAL-LINE
      AFTER ADVANCING 1 LINE

MOVE 'ASSOCIATE' TO TL-RANK.

MOVE WS-ASSOCIATE-CTR TO TL-NO-OF-EMPLOYEES .

MOVE WS-ASSOCIATE-COST TO TL-COST .

WRITE OUT-REPORT-REC FROM TL-TOTAL-LINE
      AFTER ADVANCING 1 LINE

MOVE 'ASSISTANT' TO TL-RANK.

MOVE WS-ASSISTANT-CTR TO TL-NO-OF-EMPLOYEES .

MOVE WS-ASSISTANT-COST TO TL-COST .

WRITE OUT-REPORT-REC FROM TL-TOTAL-LINE
      AFTER ADVANCING 1 LINE

MOVE 'INSTRUCTOR' TO TL-RANK.

MOVE WS-INSTRUCTOR-CTR TO TL-NO-OF-EMPLOYEES .

MOVE WS-INSTRUCTOR-COST TO TL-COST .

WRITE OUT-REPORT-REC FROM TL-TOTAL-LINE
      AFTER ADVANCING 1 LINE

ADD WS-PROFESSOR-COST , WS-ASSOCIATE-COST,
      WS-ASSISTANT-COST , WS-INSTRUCTOR-COST
      GIVING WS-TOTAL-COST.

MOVE WS-TOTAL-COST TO TL-TOTAL-COST.

WRITE OUT-REPORT-REC FROM TL-TOTAL-LINE
      AFTER ADVANCING 5 LINES.

```

78. Program to Update a sequential file

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM78.

AUTHOR. Indus Computer Services.

*PROGRAM TO UPDATE A SEQUENTIAL FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

 SELECT UPDATE-FILE ASSIGN TO DD1

 ORGANIZATION IS SEQUENTIAL

 ACCESS MODE IS SEQUENTIAL.

DATA DIVISION.

FILE SECTION.

FD UPDATE-FILE.

01 EMP-REC.

 02 EMP-NO PIC 9(3).

 02 EMP-NAME PIC X(10).

 02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

77 EMP-NO-CHANGE PIC 9(3) VALUE IS ZEROS.

77 EMP-NEW-NAME PIC X(10) VALUE IS SPACES.

77 OPTION PIC X(3) VALUE IS 'YES'.

PROCEDURE DIVISION.

 PERFORM OPEN-FILE.

 PERFORM DATA-IN.

 PERFORM UPDATE-PARA UNTIL OPTION = 'NO'.

 PERFORM CLOSE-FILE.

 STOP RUN.

OPEN-FILE.

 OPEN I-O UPDATE-FILE.

CLOSE-FILE.

 CLOSE UPDATE-FILE.

DATA-IN.

```

        DISPLAY 'ENTER THE EMPLOYEE NUMBER TO CHANGE'.
        ACCEPT EMP-NO-CHANGE.
        DISPLAY 'ENTER THE EMPLOYEE NAME TO CHANGE'.
        ACCEPT EMP-NEW-NAME.
UPDATE-PARA.
        READ UPDATE-FILE
            AT END MOVE 'NO' TO OPTION
        NOT AT END
            PERFORM PROCESS-PARA
        END-READ.
PROCESS-PARA.
        IF EMP-NO-CHANGE = EMP-NO
            MOVE EMP-NEW-NAME TO EMP-NAME
            REWRITE EMP-REC.

```

55. Program to Append a sequential file (EXTEND mode)

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM55.
AUTHOR. Indus Computer Services.
*PROGRAM TO APPEND TO A SEQUENTIAL FILE
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT APPEND-FILE ASSIGN TO DD1
            ORGANIZATION IS SEQUENTIAL
            ACCESS MODE IS SEQUENTIAL.
DATA DIVISION.
FILE SECTION.
FD APPEND-FILE.
01 EMP-REC.
        02 EMP-NO    PIC 9(3).
        02 EMP-NAME  PIC X(10).

```

```

        02 EMP-SAL    PIC 9(5).

WORKING-STORAGE SECTION.

01 IN-REC.

        02 IN-NO     PIC 9(3).

        02 IN-NAME    PIC X(10).

        02 IN-SAL     PIC 9(5).

77 OPTION    PIC X(3)  VALUE IS 'YES'.

PROCEDURE DIVISION.

        PERFORM OPEN-FILE.

        PERFORM APPEND-TO-FILE.

        PERFORM CLOSE-FILE.

        STOP RUN.

OPEN-FILE.

        OPEN EXTEND APPEND-FILE.

CLOSE-FILE.

        CLOSE APPEND-FILE.

APPEND-TO-FILE.

        PERFORM ACC-INPUT UNTIL OPTION='NO'.

ACC-INPUT.

        DISPLAY 'ENTER THE EMPLOYEE NUMBER'.

        ACCEPT IN-NO.

        DISPLAY 'ENTER THE EMPLOYEE NAME'.

        ACCEPT IN-NAME.

        DISPLAY 'ENTER THE EMPLOYEE SALARY'.

        ACCEPT IN-SAL.

        MOVE IN-REC TO EMP-REC

        WRITE EMP-REC.

        DISPLAY 'DO YOU WANT TO WRITE MORE RECORDS(YES/NO)'.

        ACCEPT OPTION.

```

79. Program to READ and to WRITE a sequential file

```

IDENTIFICATION DIVISION.

```

PROGRAM-ID. PGM79.

AUTHOR. Indus Computer Services.

*PROGRAM TO READ AND WRITE TO A SEQUENTIAL FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

 SELECT READ-FILE ASSIGN TO DD1.

 SELECT WRITE-FILE ASSIGN TO DD2.

DATA DIVISION.

FILE SECTION.

FD READ-FILE.

01 EMP-READ-REC.

 02 EMP-READ-NO PIC 9(3).

 02 EMP-READ-NAME PIC X(10).

 02 EMP-READ-SAL PIC 9(5).

FD WRITE-FILE.

01 EMP-WRITE-REC.

 02 EMP-WRITE-NO PIC 9(3).

 02 EMP-WRITE-NAME PIC X(10).

 02 EMP-WRITE-SAL PIC 9(5).

WORKING-STORAGE SECTION.

77 OPTION PIC X(3) VALUE IS 'YES'.

PROCEDURE DIVISION.

 PERFORM OPEN-FILE.

 PERFORM READ-FROM-FILE UNTIL OPTION = 'NO'.

 PERFORM CLOSE-FILE.

 STOP RUN.

OPEN-FILE.

 OPEN INPUT READ-FILE.

 OPEN OUTPUT WRITE-FILE.

CLOSE-FILE.

 CLOSE WRITE-FILE, READ-FILE.

READ-FROM-FILE.

 READ READ-FILE

```
      AT END MOVE 'NO' TO OPTION
NOT AT END
      MOVE EMP-READ-REC TO EMP-WRITE-REC
      WRITE EMP-WRITE-REC.
```

80. MATCHING LOGIC.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PGM80.
AUTHOR. Indus Computer Services.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
        SELECT OLD-MASTER ASSIGN TO DATA13O.
        SELECT TRANS-FILE ASSIGN TO DATA13T.
        SELECT NEW-MASTER ASSIGN TO DATA13N.
DATA DIVISION.
FILE SECTION.
FD OLD-MASTER
        LABEL RECORDS ARE STANDARD.
01  OLD-MASTER-REC.
        05  M-ACCT-NO                PIC      X(5) .
        05  AMOUNT-DUE                PIC      9(4)V99.
        05                                PIC      X(89) .
FD TRANS-FILE
        LABEL RECORDS ARE STANDARD.
01  TRANS-REC.
        05  T-ACCT-NO                PIC      X(5) .
        05  AMT-TRANS-IN-CURRENT-PER  PIC      9(4)V99.
        05                                PIC      X(89) .
FD NEW-MASTER
        LABEL RECORDS ARE STANDARD.
01  NEW-MASTER-REC.
```

```
05  ACCT-NO-OUT                      PIC      X(5) .
05  AMOUNT-DUE-OUT                  PIC      9(4)V99 .
05                                PIC      X(89) .
```

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

100-MAIN-MODULE.

```
    PERFORM 800-INITIALIZATION-RTN.
    PERFORM 600-READ-MASTER.
    PERFORM 700-READ-TRANS.
    PERFORM 200-COMP-RTN
        UNTIL M-ACCT-NO = HIGH-VALUES
            AND
            T-ACCT-NO  = HIGH-VALUES
    PERFORM 900-END-OF-JOB-RTN.
    STOP RUN.
```

200-COMP-RTN.

```
    EVALUATE TRUE
        WHEN T-ACCT-NO = M-ACCT-NO
            PERFORM 300-REGULAR-UPDATE
        WHEN T-ACCT-NO < M-ACCT-NO
            PERFORM 400-NEW-ACCOUNT
        WHEN OTHER
            PERFORM 500-NO-UPDATE
    END-EVALUATE.
```

300-REGULAR-UPDATE.

```
    MOVE OLD-MASTER-REC TO NEW-MASTER-REC
    COMPUTE AMOUNT-DUE-OUT = AMT-TRANS-IN-CURRENT-PER
                                + AMOUNT-DUE
    WRITE NEW-MASTER-REC
    PERFORM 600-READ-MASTER
    PERFORM 700-READ-TRANS.
```

400-NEW-ACCOUNT.

```
    MOVE SPACES TO NEW-MASTER-REC.
    MOVE T-ACCT-NO TO ACCT-NO-OUT.
```

```

        MOVE AMT-TRANS-IN-CURRENT-PER TO AMOUNT-DUE-OUT.
        WRITE NEW-MASTER-REC.
        PERFORM 700-READ-TRANS.
500-NO-UPDATE.
        WRITE NEW-MASTER-REC FROM OLD-MASTER-REC.
        PERFORM 600-READ-MASTER.
600-READ-MASTER.
        READ OLD-MASTER
            AT END
                MOVE HIGH-VALUES TO M-ACCT-NO
        END-READ.
700-READ-TRANS.
        READ TRANS-FILE
            AT END
                MOVE HIGH-VALUES TO T-ACCT-NO
        END-READ.
800-INITIALIZATION-RTN.
        OPEN INPUT OLD-MASTER
            TRANS-FILE.
        OPEN OUTPUT NEW-MASTER.
900-END-OF-JOB-RTN.
        CLOSE OLD-MASTER
            TRANS-FILE
            NEW-MASTER.

```

81. Program to SORT a sequential file (ASCENDING)

```

IDENTIFICATION DIVISION.
PROGRAM-ID. PGM64.
AUTHOR. Indus Computer Services.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

```

```

        SELECT UNSORT-FILE ASSIGN TO DD1.

        SELECT WORK-FILE ASSIGN TO DD2.

        SELECT SORT-FILE ASSIGN TO DD3.

DATA DIVISION.

FILE SECTION.

FD UNSORT-FILE.

01 UNSORT-REC.

    02 UNSORT-EMP-NO    PIC 9(3).

    02 UNSORT-EMP-NAME  PIC X(10).

    02 UNSORT-EMP-SAL   PIC 9(5).

FD SORT-FILE.

01 SORT-REC.

    02 SORT-EMP-NO    PIC 9(3).

    02 SORT-EMP-NAME  PIC X(10).

    02 SORT-EMP-SAL   PIC 9(5).

SD WORK-FILE.

01 WORK-REC.

    02 WORK-EMP-NO    PIC 9(3).

    02 WORK-EMP-NAME  PIC X(10).

    02 WORK-EMP-SAL   PIC 9(5).

PROCEDURE DIVISION.

        SORT WORK-FILE ON ASCENDING KEY WORK-EMP-NO

                USING UNSORT-FILE

                GIVING SORT-FILE.

        STOP RUN.

```

82. Program to SORT a sequential file (DESCENDING)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM82.

AUTHOR. Indus Computer Services.

*PROGRAM TO SORT A FILE (DESCENDING)

ENVIRONMENT DIVISION.

```

```

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT UNSORT-FILE ASSIGN TO DD1.

    SELECT WORK-FILE ASSIGN TO DD2.

    SELECT SORT-FILE ASSIGN TO DD3.

DATA DIVISION.

FILE SECTION.

FD UNSORT-FILE.

01 UNSORT-REC.

    02 UNSORT-EMP-NO    PIC 9(3).

    02 UNSORT-EMP-NAME  PIC X(10).

    02 UNSORT-EMP-SAL   PIC 9(5).

FD SORT-FILE.

01 SORT-REC.

    02 SORT-EMP-NO    PIC 9(3).

    02 SORT-EMP-NAME  PIC X(10).

    02 SORT-EMP-SAL   PIC 9(5).

SD WORK-FILE.

01 WORK-REC.

    02 WORK-EMP-NO    PIC 9(3).

    02 WORK-EMP-NAME  PIC X(10).

    02 WORK-EMP-SAL   PIC 9(5).

PROCEDURE DIVISION.

    SORT WORK-FILE ON DESCENDING KEY WORK-EMP-NO

        USING UNSORT-FILE

        GIVING SORT-FILE.

    STOP RUN.

```

83. Program to MERGE a sequential file

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM83.

AUTHOR. Indus Computer Services.

```

```
*PROGRAM TO MERGE FILES

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT FILE1 ASSIGN TO DD1.

    SELECT FILE2 ASSIGN TO DD2.

    SELECT WORK-FILE ASSIGN TO DD3.

    SELECT MERGE-FILE ASSIGN TO DD4.

DATA DIVISION.

FILE SECTION.

FD FILE1.

01 FILE1-REC.

    02 FILE1-EMP-NO      PIC 9(3).

    02 FILE1-EMP-NAME    PIC X(10).

    02 FILE1-EMP-SAL     PIC 9(5).

FD FILE2.

01 FILE2-REC.

    02 FILE2-EMP-NO      PIC 9(3).

    02 FILE2-EMP-NAME    PIC X(10).

    02 FILE2-EMP-SAL     PIC 9(5).

FD MERGE-FILE.

01 MERGE-REC.

    02 MERGE-EMP-NO      PIC 9(3).

    02 MERGE-EMP-NAME    PIC X(10).

    02 MERGE-EMP-SAL     PIC 9(5).

SD WORK-FILE.

01 WORK-REC.

    02 WORK-EMP-NO       PIC 9(3).

    02 WORK-EMP-NAME     PIC X(10).

    02 WORK-EMP-SAL      PIC 9(5).

PROCEDURE DIVISION.

    MERGE WORK-FILE ON ASCENDING KEY WORK-EMP-NO

        USING FILE1,FILE2

        GIVING MERGE-FILE.
```

STOP RUN.

84. Program to SORT a sequential file with INPUT PROCEDURE

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM84.

AUTHOR. Indus Computer Services.

*PROGRAM TO SORT A RECORD WITH SAL < 3000(INPUT PROCEDURE)

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT UNSORT-FILE ASSIGN TO DD1.

SELECT WORK-FILE ASSIGN TO DD2.

SELECT SORT-FILE ASSIGN TO DD3.

DATA DIVISION.

FILE SECTION.

FD UNSORT-FILE.

01 UNSORT-REC.

02 UNSORT-EMP-NO PIC 9(3).

02 UNSORT-EMP-NAME PIC X(10).

02 UNSORT-EMP-SAL PIC 9(5).

FD SORT-FILE.

01 SORT-REC.

02 SORT-EMP-NO PIC 9(3).

02 SORT-EMP-NAME PIC X(10).

02 SORT-EMP-SAL PIC 9(5).

SD WORK-FILE.

01 WORK-REC.

02 WORK-EMP-NO PIC 9(3).

02 WORK-EMP-NAME PIC X(10).

02 WORK-EMP-SAL PIC 9(5).

PROCEDURE DIVISION.

SORT WORK-FILE ON ASCENDING KEY WORK-EMP-NO

```

        INPUT PROCEDURE IS CHK-SAL
        GIVING SORT-FILE.

    STOP RUN.

CHK-SAL SECTION.

PARA-OPEN.

    OPEN INPUT UNSORT-FILE.

PARA-READ.

    READ UNSORT-FILE AT END GO TO PARA-CLOSE.

    IF UNSORT-EMP-SAL < 3000

        RELEASE WORK-REC FROM UNSORT-REC

        GO TO PARA-READ

    ELSE

        GO TO PARA-READ.

PARA-CLOSE.

    CLOSE UNSORT-FILE.

```

85. Program to SORT a sequential file with OUTPUT PROCEDURE

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM85.

AUTHOR. Indus Computer Services.

*PROGRAM TO SORT A RECORD WITH SAL < 3000 (OUTPUT PROCEDURE)

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT UNSORT-FILE ASSIGN TO DD1.

    SELECT WORK-FILE ASSIGN TO DD2.

    SELECT SORT-FILE ASSIGN TO DD3.

DATA DIVISION.

FILE SECTION.

FD UNSORT-FILE.

01 UNSORT-REC.

```

```

    02 UNSORT-EMP-NO      PIC 9(3) .
    02 UNSORT-EMP-NAME    PIC X(10) .
    02 UNSORT-EMP-SAL     PIC 9(5) .

FD SORT-FILE.

01 SORT-REC.
    02 SORT-EMP-NO      PIC 9(3) .
    02 SORT-EMP-NAME    PIC X(10) .
    02 SORT-EMP-SAL     PIC 9(5) .

SD WORK-FILE.

01 WORK-REC.
    02 WORK-EMP-NO      PIC 9(3) .
    02 WORK-EMP-NAME    PIC X(10) .
    02 WORK-EMP-SAL     PIC 9(5) .

PROCEDURE DIVISION.

    SORT WORK-FILE ON ASCENDING KEY WORK-EMP-NO
        USING UNSORT-FILE
        OUTPUT PROCEDURE IS CHK-SAL.

    STOP RUN.

CHK-SAL SECTION.

    PARA-OPEN.

        OPEN OUTPUT SORT-FILE.

    PARA-READ.

        RETURN WORK-FILE RECORD INTO SORT-REC
            AT END GO TO PARA-CLOSE.

        IF SORT-EMP-SAL < 3000
            WRITE SORT-REC
            GO TO PARA-READ
        ELSE
            GO TO PARA-READ.

    PARA-CLOSE.

        CLOSE SORT-FILE.

```

86. Program to write to an indexed file.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM86.

AUTHOR. Indus Computer Services.

*PROGRAM TO WRITE TO A INDEXED FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

 SELECT WRITE-FILE ASSIGN TO DD1

 ORGANIZATION IS INDEXED

 ACCESS MODE IS RANDOM

 RECORD KEY IS EMP-NO.

DATA DIVISION.

FILE SECTION.

FD WRITE-FILE.

01 EMP-REC.

 02 EMP-NO PIC 9(3).

 02 EMP-NAME PIC X(10).

 02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

01 IN-REC.

 02 IN-NO PIC 9(3).

 02 IN-NAME PIC X(10).

 02 IN-SAL PIC 9(5).

77 OPTION PIC X(3) VALUE IS 'YES'.

77 I PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

 PERFORM OPEN-FILE.

 PERFORM WRITE-TO-FILE.

 PERFORM CLOSE-FILE.

 STOP RUN.

OPEN-FILE.

 OPEN OUTPUT WRITE-FILE.

CLOSE-FILE.

```

        CLOSE WRITE-FILE.

WRITE-TO-FILE.

        PERFORM ACC-INPUT UNTIL OPTION='NO'.

ACC-INPUT.

        COMPUTE I = I + 1.

        DISPLAY 'ENTER THE EMPLOYEE NUMBER'.

        ACCEPT IN-NO.

        DISPLAY 'ENTER THE EMPLOYEE NAME'.

        ACCEPT IN-NAME.

        DISPLAY 'ENTER THE EMPLOYEE SALARY'.

        ACCEPT IN-SAL.

        MOVE IN-REC TO EMP-REC

        WRITE EMP-REC.

        DISPLAY 'DO YOU WANT TO WRITE MORE RECORDS (YES/NO)'.

        ACCEPT OPTION.

```

87. Program to Read from an indexed file.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM87.

AUTHOR. Indus Computer Services.

*PROGRAM TO READ FROM A INDEXED FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

        SELECT READ-FILE ASSIGN TO DD1

                ORGANIZATION IS INDEXED

                ACCESS MODE IS RANDOM

                RECORD KEY IS EMP-NO.

DATA DIVISION.

FILE SECTION.

FD READ-FILE.

01 EMP-REC.

```

```

02 EMP-NO      PIC 9(3).
02 EMP-NAME    PIC X(10).
02 EMP-SAL     PIC 9(5).
WORKING-STORAGE SECTION.
01 OUT-REC.
    02 OUT-NO      PIC 9(3).
    02 OUT-NAME    PIC X(10).
    02 OUT-SAL     PIC 9(5).
77 EMP-NUMBER   PIC 9(3)  VALUE IS ZERO.
PROCEDURE DIVISION.
    PERFORM OPEN-FILE.
    PERFORM ACC-INPUT.
    PERFORM READ-PARA.
    PERFORM CLOSE-FILE.
    STOP RUN.
OPEN-FILE.
    OPEN INPUT READ-FILE.
CLOSE-FILE.
    CLOSE READ-FILE.
ACC-INPUT.
    DISPLAY 'ENTER THE EMPLOYEE NUMBER TO FIND'.
    ACCEPT EMP-NUMBER.
READ-PARA.
    MOVE EMP-NUMBER TO EMP-NO.
    READ READ-FILE RECORD
        KEY IS EMP-NO
        INVALID KEY DISPLAY 'RECORD NOT FOUND'.
    MOVE EMP-REC TO OUT-REC.
    DISPLAY OUT-REC.
    STOP RUN.

```

88. Program to update to an indexed file.

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM88.

AUTHOR. Indus Computer Services.

*PROGRAM TO UPDATE AN INDEXED FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT UPDATE-FILE ASSIGN TO DD1

ORGANIZATION IS INDEXED

ACCESS MODE IS RANDOM

RECORD KEY IS EMP-NO.

DATA DIVISION.

FILE SECTION.

FD UPDATE-FILE.

01 EMP-REC.

02 EMP-NO PIC 9(3).

02 EMP-NAME PIC X(10).

02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

77 EMP-NEW-NO PIC 9(3) VALUE IS ZEROS.

77 EMP-NEW-NAME PIC X(10) VALUE IS SPACES.

77 EMP-NEW-SALARY PIC 9(5) VALUE IS ZEROS.

77 OPTION PIC X(3) VALUE IS 'YES'.

77 EMP-NUMBER PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

PERFORM OPEN-FILE.

PERFORM DATA-IN.

PERFORM UPDATE-PARA.

PERFORM CLOSE-FILE.

STOP RUN.

OPEN-FILE.

OPEN I-O UPDATE-FILE.

CLOSE-FILE.

CLOSE UPDATE-FILE.

```

DATA-IN.

    DISPLAY 'ENTER THE EMP NUMBER TO UPDATE'.

    ACCEPT EMP-NUMBER.

    DISPLAY 'ENTER THE NEW NAME'.

    ACCEPT EMP-NEW-NAME.

    DISPLAY 'ENTER THE NEW SALARY'.

    ACCEPT EMP-NEW-SALARY.

UPDATE-PARA.

    MOVE EMP-NUMBER TO EMP-NO.

    MOVE EMP-NEW-NAME TO EMP-NAME.

    MOVE EMP-NEW-SALARY TO EMP-SAL.

    REWRITE EMP-REC

        INVALID KEY DISPLAY 'RECORD NOT FOUND'.

```

89. Program to DELETE from an indexed file.

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM89.

AUTHOR. Indus Computer Services.

*PROGRAM TO DELETE FROM A INDEXED FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT DELETE-FILE ASSIGN TO DD1

        ORGANIZATION IS INDEXED

        ACCESS MODE IS RANDOM

        RECORD KEY IS EMP-NO.

DATA DIVISION.

FILE SECTION.

FD DELETE-FILE.

01 EMP-REC.

    02 EMP-NO    PIC 9(3).

    02 EMP-NAME  PIC X(10).

```

```

        02 EMP-SAL      PIC 9(5) .

WORKING-STORAGE SECTION.

77 EMP-NUMBER  PIC 9(3)  VALUE IS ZERO.

PROCEDURE DIVISION.

        PERFORM OPEN-FILE.

        PERFORM DATA-IN.

        PERFORM DELETE-PARA.

        PERFORM CLOSE-FILE.

        STOP RUN.

OPEN-FILE.

        OPEN I-O DELETE-FILE.

CLOSE-FILE.

        CLOSE DELETE-FILE.

DATA-IN.

        DISPLAY 'ENTER THE EMPLOYEE NUMBER TO DELETE'.

        ACCEPT EMP-NUMBER.

DELETE-PARA.

        MOVE EMP-NUMBER TO EMP-NO.

        DELETE DELETE-FILE RECORD

                INVALID KEY DISPLAY 'RECORD NOT FOUND'.

```

90. Program to READ from a relative file

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM90.

AUTHOR. Indus Computer Services.

*PROGRAM TO READ FROM A RELATIVE FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

        SELECT READ-FILE ASSIGN TO DD1

                ORGANIZATION IS RELATIVE

                ACCESS MODE IS RANDOM,RELATIVE KEY IS REL-POSITION.

```

DATA DIVISION.

FILE SECTION.

FD READ-FILE.

01 EMP-REC.

02 EMP-NO PIC 9(3).

02 EMP-NAME PIC X(10).

02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

01 OUT-REC.

02 OUT-NO PIC 9(3).

02 OUT-NAME PIC X(10).

02 OUT-SAL PIC 9(5).

77 OPTION PIC X(3) VALUE IS 'YES'.

77 REL-POSITION PIC 9(3) VALUE IS ZERO.

77 I PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

PERFORM OPEN-FILE.

DISPLAY 'ENTER THE RELATIVE POSITION'.

ACCEPT I.

PERFORM READ-PARA.

PERFORM CLOSE-FILE.

STOP RUN.

OPEN-FILE.

OPEN INPUT READ-FILE.

CLOSE-FILE.

CLOSE READ-FILE.

READ-PARA.

MOVE I TO REL-POSITION.

READ READ-FILE RECORD

INVALID KEY DISPLAY 'RECORD NOT FOUND'.

MOVE EMP-REC TO OUT-REC.

DISPLAY OUT-REC.

STOP RUN.

91. Program to WRITE a relative file

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM91.

AUTHOR. Indus Computer Services.

*PROGRAM TO WRITE TO A RELATIVE FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT WRITE-FILE ASSIGN TO DD1

ORGANIZATION IS RELATIVE

ACCESS MODE IS RANDOM,RELATIVE KEY IS REL-POSITION.

DATA DIVISION.

FILE SECTION.

FD WRITE-FILE.

01 EMP-REC.

02 EMP-NO PIC 9(3).

02 EMP-NAME PIC X(10).

02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

01 IN-REC.

02 IN-NO PIC 9(3).

02 IN-NAME PIC X(10).

02 IN-SAL PIC 9(5).

77 OPTION PIC X(3) VALUE IS 'YES'.

77 REL-POSITION PIC 9(3) VALUE IS ZERO.

77 I PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

PERFORM OPEN-FILE.

PERFORM WRITE-TO-FILE.

PERFORM CLOSE-FILE.

STOP RUN.

OPEN-FILE.

```

        OPEN OUTPUT WRITE-FILE.

CLOSE-FILE.

        CLOSE WRITE-FILE.

WRITE-TO-FILE.

        PERFORM ACC-INPUT UNTIL OPTION='NO'.
ACC-INPUT.

        COMPUTE I = I + 1.

        DISPLAY 'ENTER THE EMPLOYEE NUMBER'.

        ACCEPT IN-NO.

        DISPLAY 'ENTER THE EMPLOYEE NAME'.

        ACCEPT IN-NAME.

        DISPLAY 'ENTER THE EMPLOYEE SALARY'.

        ACCEPT IN-SAL.

        MOVE IN-REC TO EMP-REC

        MOVE I TO REL-POSITION.

        WRITE EMP-REC.

        DISPLAY 'DO YOU WANT TO WRITE MORE RECORDS (YES/NO)'.

        ACCEPT OPTION.

```

92. Program to UPDATE a relative file

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM92.

AUTHOR. Indus Computer Services.

*PROGRAM TO UPDATE A RRELATIVE FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

        SELECT UPDATE-FILE ASSIGN TO DD1

                ORGANIZATION IS RELATIVE

                ACCESS MODE IS RANDOM, RELATIVE KEY IS REL-POSITION.

DATA DIVISION.

FILE SECTION.

```

FD UPDATE-FILE.

01 EMP-REC.

02 EMP-NO PIC 9(3).

02 EMP-NAME PIC X(10).

02 EMP-SAL PIC 9(5).

WORKING-STORAGE SECTION.

77 EMP-NEW-NO PIC 9(3) VALUE IS ZEROS.

77 EMP-NEW-NAME PIC X(10) VALUE IS SPACES.

77 EMP-NEW-SALARY PIC 9(5) VALUE IS ZEROS.

77 OPTION PIC X(3) VALUE IS 'YES'.

77 REL-POSITION PIC 9(3) VALUE IS ZERO.

77 I PIC 9(3) VALUE IS ZERO.

PROCEDURE DIVISION.

PERFORM OPEN-FILE.

PERFORM DATA-IN.

PERFORM UPDATE-PARA.

PERFORM CLOSE-FILE.

STOP RUN.

OPEN-FILE.

OPEN I-O UPDATE-FILE.

CLOSE-FILE.

CLOSE UPDATE-FILE.

DATA-IN.

DISPLAY 'ENTER THE NEW NO'.

ACCEPT EMP-NEW-NO.

DISPLAY 'ENTER THE NEW NAME'.

ACCEPT EMP-NEW-NAME.

DISPLAY 'ENTER THE NEW SALARY'.

ACCEPT EMP-NEW-SALARY.

DISPLAY 'ENTER THE RELATIVE POSITION'.

ACCEPT I.

UPDATE-PARA.

MOVE EMP-NEW-NAME TO EMP-NAME.

MOVE EMP-NEW-NO TO EMP-NO.

```
MOVE EMP-NEW-SALARY TO EMP-SAL.

MOVE I TO REL-POSITION.

REWRITE EMP-REC

INVALID KEY DISPLAY 'RECORD NOT FOUND'.
```

93. Program to DELETE a relative file

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM93.

AUTHOR. Indus Computer Services.

*PROGRAM TO DELETE FROM A RRELATIVE FILE

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

    SELECT DELETE-FILE ASSIGN TO DD1

    ORGANIZATION IS RELATIVE

    ACCESS MODE IS RANDOM,RELATIVE KEY IS REL-POSITION.

DATA DIVISION.

FILE SECTION.

FD DELETE-FILE.

01 EMP-REC.

    02 EMP-NO    PIC 9(3).

    02 EMP-NAME  PIC X(10).

    02 EMP-SAL   PIC 9(5).

WORKING-STORAGE SECTION.

77 REL-POSITION PIC 9(3)  VALUE IS ZERO.

77 I          PIC 9(3)  VALUE IS ZERO.

PROCEDURE DIVISION.

    PERFORM OPEN-FILE.

    PERFORM DATA-IN.

    PERFORM DELETE-PARA.

    PERFORM CLOSE-FILE.

    STOP RUN.
```

```

OPEN-FILE.

    OPEN I-O DELETE-FILE.

CLOSE-FILE.

    CLOSE DELETE-FILE.

DATA-IN.

    DISPLAY 'ENTER THE RELATIVE POSITION TO DELETE'.

    ACCEPT I.

DELETE-PARA.

    MOVE I TO REL-POSITION.

    DELETE DELETE-FILE RECORD

        INVALID KEY DISPLAY 'RECORD NOT FOUND'.

```

94. Program to INSPECT A STRING (NO OF OCCURANCE OF A CHARACTER)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM94.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING(NO OF OCCURANCE OF A CHAR)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

77 NOOFOCCUR PIC 9 VALUE IS ZERO.

PROCEDURE DIVISION.

    INSPECT INSTRING TALLYING NOOFOCCUR FOR ALL 'S'.

    DISPLAY NOOFOCCUR.

    STOP RUN.

```

95. Program to INSPECT A STRING (NO OF OCCURANCE OF A CHARACTER BEFORE A CHARACTER)

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM95.

```

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING

*(NO OF OCCURANCE OF A CHAR BEFORE A CHAR)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

77 NOOFOCCUR PIC 9 VALUE IS ZERO.

PROCEDURE DIVISION.

INSPECT INSTRING TALLYING NOOFOCCUR FOR ALL 'S' BEFORE
INITIAL 'A'.

DISPLAY NOOFOCCUR.

STOP RUN.

96. Program to INSPECT A STRING (NO OF OCCURANCE OF A CHARACTER AFTER A CHARACTER)

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM96.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING

*(NO OF OCCURANCE OF A CHAR AFTER A CHAR)

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

77 NOOFOCCUR PIC 9 VALUE IS ZERO.

PROCEDURE DIVISION.

INSPECT INSTRING TALLYING NOOFOCCUR FOR ALL 'S' AFTER
INITIAL 'A'.

DISPLAY NOOFOCCUR.

STOP RUN.

97. Program to INSPECT A STRING AND TO REPLACE A CHARACTER

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM97.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING AND TO REPLACE
*A CHAR

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

PROCEDURE DIVISION.

    INSPECT INSTRING REPLACING ALL 'S' BY 's'.

    DISPLAY INSTRING.

    STOP RUN.
```

98. Program to INSPECT A STRING AND TO REPLACE A CHARACTER AFTER A CHARACTER

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM98.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING AND TO REPLACE
*A CHAR AFTER A CHARACTER.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

PROCEDURE DIVISION.

    INSPECT INSTRING REPLACING ALL 'S' BY 's' AFTER INITIAL 'A'.

    DISPLAY INSTRING.

    STOP RUN.
```

99. Program to INSPECT A STRING AND TO REPLACE A CHARACTER BEFORE A CHARACTER

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM99.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING AND TO REPLACE

*A CHAR BEFORE A CHARACTER.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

PROCEDURE DIVISION.

INSPECT INSTRING REPLACING ALL 'S' BY 's' BEFORE INITIAL 'A'.

DISPLAY INSTRING.

STOP RUN.

100. Program to INSPECT A STRING TO FIND THE NO OF OCCURANCE
AND REPLACE A CHARACTER

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM100.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING

*TO FIND NO OF OCCURANCE AND TO REPLACE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(30) VALUE IS 'INDUS CONSULTANT SERVICES'.

77 NOOFOCCUR PIC 9 VALUE IS ZERO.

PROCEDURE DIVISION.

INSPECT INSTRING TALLYING NOOFOCCUR FOR ALL 'S'

REPLACING ALL 'S' BY 's'.

DISPLAY NOOFOCCUR.

DISPLAY INSTRING.

STOP RUN.

101. Program to INSPECT THE NUMBER OF CHARACTERS AFTER A CHARACTER

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM101.

AUTHOR. Indus Computer Services.

*PROGRAM TO INSPECT A STRING

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 INSTRING PIC X(25) VALUE IS 'INDUS CONSULTANT SERVICES'.

77 NOOFOCCUR PIC 99 VALUE IS ZERO.

PROCEDURE DIVISION.

INSPECT INSTRING TALLYING NOOFOCCUR FOR CHARACTERS AFTER
INITIAL 'S'.

DISPLAY NOOFOCCUR.

STOP RUN.

102. Program to CONCATENATE STRINGS : STRING

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM102.

AUTHOR. Indus Computer Services.

*PROGRAM TO CONCATENATE A STRING WITH ANOTHER STRING.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 STRING1 PIC X(6) VALUE IS 'INDUS'.

77 STRING2 PIC X(11) VALUE IS 'CONSULTANT '.

77 STRING3 PIC X(8) VALUE IS 'SERVICES'.

```

77 STRING4    PIC X(30)  VALUE IS SPACES.

PROCEDURE DIVISION.

    STRING STRING1,STRING2,STRING3

        DELIMITED BY SIZE INTO STRING4.

    DISPLAY STRING4.

    STOP RUN.

```

103. Program to SPLIT STRINGS : UNSTRING

```

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM103.

AUTHOR. Indus Computer Services.

*PROGRAM TO SPLIT A STRING INTO STRINGS BASED ON DELIMITOR.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 STRING1    PIC X(30)  VALUE IS

                                                    'INDUS
CONSULTANT SERVICES'.

77 STRING2    PIC X(5)   VALUE IS SPACES.

77 STRING3    PIC X(10)  VALUE IS SPACES.

77 STRING4    PIC X(8)   VALUE IS SPACES.

PROCEDURE DIVISION.

    UNSTRING STRING1

        DELIMITED BY ALL SPACES INTO STRING2,STRING3,STRING4.

    DISPLAY STRING2.

    DISPLAY STRING3.

    DISPLAY STRING4.

    STOP RUN.

```

104. Program to CALL SUBROUTINES

```

IDENTIFICATION DIVISION.

```

```
PROGRAM-ID. PGM104.

AUTHOR. Indus Computer Services.

*PROGRAM TO CALL SUBROUTINES

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

    CALL 'SUBPGM'.

    STOP RUN.
```

SUBPGM

```
IDENTIFICATION DIVISION.

PROGRAM-ID. SUBPGM.

AUTHOR. Indus Computer Services.

*SUB PROGRAM

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

    DISPLAY 'HAI'.

    EXIT PROGRAM.
```

105. Program to CALL SUBROUTINES AND PASSING VALUE BY REFERENCE

```
IDENTIFICATION DIVISION.

PROGRAM-ID. PGM105.

AUTHOR. Indus Computer Services.

*PROGRAM TO CALL SUBROUTINES AND PASSING VALUE BY REFERENCE

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.
```

```
77 A  PIC 9  VALUE IS 3.
77 B  PIC 9  VALUE IS 6.
PROCEDURE DIVISION.
    CALL 'SUBPGM1' USING A B.
    DISPLAY B.
    STOP RUN.
```

SUBPGM1

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SUBPGM1.
AUTHOR. Indus Computer Services.
*SUB PROGRAM
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
77 A  PIC 9.
77 B  PIC 9.
PROCEDURE DIVISION USING A B.
    ADD A TO B.
    EXIT PROGRAM.
```

106. Program to CALL SUBROUTINES AND PASSING VALUE BY VALUE

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PGM106.
AUTHOR. Indus Computer Services.
*PROGRAM TO CALL SUBROUTINES AND PASSING VALUE BY VALUE
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 A  PIC 9  VALUE IS 3.
77 B  PIC 9  VALUE IS 6.
```

PROCEDURE DIVISION.

CALL 'SUBPGM1' USING BY CONTENT A B.

DISPLAY B.

STOP RUN.

SUBPGM1

IDENTIFICATION DIVISION.

PROGRAM-ID. SUBPGM1.

AUTHOR. Indus Computer Services.

*SUB PROGRAM

ENVIRONMENT DIVISION.

DATA DIVISION.

LINKAGE SECTION.

77 A PIC 9.

77 B PIC 9.

PROCEDURE DIVISION USING A B.

ADD A TO B.

EXIT PROGRAM.

107. Program to CALL SUBROUTINES DYNAMICALLY

IDENTIFICATION DIVISION.

PROGRAM-ID. PGM107.

AUTHOR. Indus Computer Services.

*PROGRAM TO CALL SUBROUTINES DYNAMICALLY

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

77 SUBNAME PIC X(6) VALUE IS SPACES.

PROCEDURE DIVISION.

MOVE 'SUBPGM' TO SUBNAME.

CALL SUBNAME

STOP RUN.

SUBPGM

IDENTIFICATION DIVISION.

PROGRAM-ID. SUBPGM.

AUTHOR. Indus Computer Services.

*SUB PROGRAM

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

PROCEDURE DIVISION.

DISPLAY 'HAI'.

EXIT PROGRAM.

108. program for DANAMIC CALL PROGRAMS:

IDENTIFICATION DIVISION.

PROGRAM-ID. Pgm108.

AUTHOR. Indus Computer Services.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 WS-REC.

05 A PIC 99.

05 B PIC 99.

05 C PIC 9999.

77 WS-REC-1 PIC X(6) VALUE 'PGM83A'.

77 WS-REC-2 PIC X(6) VALUE 'PGM83B'.

LINKAGE SECTION.

PROCEDURE DIVISION.

MAIN-PARA.

```
ACCEPT A.  
ACCEPT B.  
CALL WS-REC-1 USING BY REFERENCE A, B, C.  
DISPLAY 'C= ' C.  
CALL WS-REC-2 USING BY REFERENCE A, B,C.  
DISPLAY 'C= ' C.  
STOP RUN.
```

Sub Program PGM83A:

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM83A.  
AUTHOR. Indus Computer Services.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
LINKAGE SECTION.  
01 X PIC 99.  
01 Y PIC 99.  
01 Z PIC 9999.  
PROCEDURE DIVISION USING X, Y, Z.  
    COMPUTE Z = X + Y.  
    EXIT PROGRAM.
```

PGM83B:

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. PGM83B.  
AUTHOR. Indus Computer Services.  
ENVIRONMENT DIVISION.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
LINKAGE SECTION.  
01 X PIC 99.
```

```

01 Y      PIC 99.
01 Z      PIC 9999.
PROCEDURE DIVISION USING X, Y, Z.
    COMPUTE Z = X * Y.
    EXIT PROGRAM.

```

COBOL program to print Even and Odd Numbers between range using IF Condition.

PROGRAM:

```

IDENTIFICATION DIVISION.
PROGRAM-ID P2A.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
    01 NUM1 PIC 9(3).
    01 NUM2 PIC 9(3).
    01 R PIC 9(2).
    01 N PIC 9(2).
    01 I PIC 9(2).
    01 J PIC 9(2).
    01 M PIC Z(3).
PROCEDURE DIVISION.
DISPLAY-PARA.
    DISPLAY 'ENTER THE LOWER LIMIT'.
    DISPLAY '-----'.
    ACCEPT NUM1.
    DISPLAY 'ENTER THE UPPER LIMIT'.
    DISPLAY '-----'.
    ACCEPT NUM2.
    IF NUM1 > NUM2
        DISPLAY 'INVALID INPUT'
    STOP RUN.
    DISPLAY '-----'.

```

```

    DISPLAY 'LIST OF EVEN NUMBERS'.
    DISPLAY '-----'.
    PERFORM CAL-PARA VARYING I FROM NUM1 BY 1 UNTIL I>NUM2.
    DISPLAY '-----'.
    DISPLAY 'LIST OF ODD NUMBERS'.
    DISPLAY '-----'.
    PERFORM CAL1-PARA VARYING I FROM NUM1 BY 1 UNTIL I>NUM2.
    STOP RUN.

CAL-PARA.
    DIVIDE I BY 2 GIVING N REMAINDER R.
    IF R = 0
    MOVE I TO J
    DISPLAY SPACES " | " J " | ".

CAL1-PARA.
    DIVIDE I BY 2 GIVING N REMAINDER R.
    IF R NOT = 0
    MOVE I TO M
    DISPLAY SPACES " | " M " | ".

```